In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

University of Alberta

A New Bandwidth Allocation Scheme for 3G Environment

by

Yuan Sha

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Fall 2003

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisisitons et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-87881-3
Our file Notre référence
ISBN: 0-612-87881-3

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

Canada

University of Alberta

Library Release Form

Name of Author: Yuan Sha

Title of Thesis: A New Bandwidth Allocation Scheme for 3G Environment

Degree: Master of Science

Year this Degree Granted: 2003

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Date: Aug. 8, 2003

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **A New Bandwidth Allocation Scheme for 3G Environment** submitted by Yuan Sha in partial fulfillment of the requirements for the degree of **Master of Science** in *Communication Networks*.

Date: Aug. 1, 2003

Abstract

A novel scheme is proposed in this thesis to enhance the bandwidth utilization and increase admission ratio of handoff and new users in the Third Generation Wireless Networks (3G) environments. To achieve this goal, we design a new bandwidth allocation algorithm which efficiently assigns bandwidth to mobile users, and a bandwidth degradation algorithm that enables more handoff users admitted if the bandwidth is exhausted. A crucial notion used in both algorithms is the *User Profile*, which uses the least data to record the Quality of Service (QoS) characteristics of every connection of each user. Each Base Station (BS) maintains a *User Profile* for all alive users in its coverage cell. During the handoff, the QoS information related to a handoff user in the *User Profile* is forwarded from the source BS to the target BS via the Mobile Switching Center (MSC). After acquiring the QoS requirement of the handoff user at the level of application, the target BS is able to accordingly allocate bandwidth with benefits of efficient bandwidth utilization and constant service quality experienced by mobile user.

Differentiating the real time connection and non-real time connection makes possible our bandwidth degradation algorithm. The high delay tolerance of non-real time connections enables the BS to *temporarily* take away the bandwidth from a user to accommodate handoff and new users. Therefore, the admission ratio is highly improved by chopping the bandwidth assigned to non-real time connections without much service degradation.

Both the allocation algorithm and the degradation algorithm are application-aware, class-based, and dynamic, with the help of *User Profile*. To assess the new proposed scheme, we develop a simulation system and compare its performance with the conven-

tional approach. Our result shows that the performance is improved greatly in the aspects of admission ratio and resources utilization.

To the love from my parents, my sister, and my wife,
To the guide from my supervisor, Dr. Janelle Harms,
To the friendship from pals,
To the people who helped me before

Acknowledgments

I thank my parents and my sister for their love and long-time encouragement. I would thank my wife Qiong Xie for her encouragement and technical advice in aspect of wireless communications.

I would like to express my sincere appreciation to my supervisor Dr. Janelle Harms for her insightful guidance, exceptional expertise, continuous encouragement and financial support during this work. I would greatly thank Dr. Herb Yang for generously allowing me to access some resources in his lab.

Special thanks to my friends in our Communications Network Group Lab and other labs for their helps and unforgettable friendships.

Thanks to all the people who ever helped me.

This thesis was financially supported through research assistantships provided by the Canadian Institute Telecommunication Research (CITR).

Contents

1 Introduction 1

1.1 From CDMA to 3G 1

1.1.1 CDMA 2

1.1.2 3G 3

1.1.3 Architecture 5

1.2 Handoffs Mechanisms in 3G 6

1.3 Thesis Outline and Contributions 6

2 Literature Review 8

2.1 Bandwidth Allocation Mechanisms Dealing with Handoffs 9

2.1.1 DiffServ and Its Variants 9

2.1.2 Guarded Channel Scheme and Its Variants 10

2.1.3 RSVP and Its Variants 13

2.1.4 Admission Control 14

2.2 Summary 14

3 The Bandwidth Allocation Mechanism 16

3.1 User Profile 18

3.1.1 The Structure of The User Profile 18

3.1.2 Establishing the User Profile 23

3.1.3 How to Apply the User Profile to The Handoff User 26

3.1.4 Summary 30

3.2 Bandwidth Allocation Mechanism 31
3.2.1 Bandwidth Allocation Algorithm 32
3.2.2 Bandwidth Degradation Algorithm 41
3.2.3 Summary and Discussion 47
3.3 The Entire Working Process of the Source BS 50
3.4 Summary 53

4 Simulation Systems **55**
4.1 Simulation 56
4.1.1 Simulation Modelling 56
4.2 Development Environment 64
4.2.1 Microsoft .NET Framework and C# 65
4.2.2 The Interface: The Place to Input Parameters 67
4.2.3 The Simulation Logic 67
4.2.4 Data Storage: Plain Text or Database 69
4.3 Verification and Validation 71
4.3.1 Verification 71
4.3.2 Validation 73
4.4 Summary 73

5 Simulation Results and Analysis **75**
5.1 Performance Metrics 76
5.2 Assumptions and Parameters 77
5.2.1 Assumptions 77
5.2.2 Parameters 79
5.2.3 Parameters Required by ThesisBaseCase 79
5.2.4 Parameters Required by ThesisMyCase 81
5.2.5 The format of ParamtersFile.txt 82

5.3 Performance Evaluation 84
5.3.1 Experiments 85
5.3.2 Transient Removal 85
5.3.3 Performance Comparison 86
5.4 Summary 101

6 Conclusions and Future Works **106**
6.1 Contributions 106
6.2 Conclusions 107
6.3 Future Work 109

Bibliography **110**

List of Figures

1.1 The Relationship among MS, BS and MSC 5

3.1 A Typical Example of Forwarding User Profile 27

3.2 The Finite State Machine for A Working Source BS 50

4.1 An Example of the Working Process of the Simulation 61

4.2 ThesisBaseCase: Users Can Input Needed Parameters In Text Boxes 67

4.3 ThesisBaseCase: Users Can Input Needed Parameters From An Existing Parameter File 68

4.4 ThesisMyCase: Users Can Input Needed Parameters In Text Boxes 68

4.5 ThesisMyCase: Users Can Input Needed Parameters From An Existing Parameter File 69

4.6 ThesisBaseCase: Users Can Specify The Folder For Log Files 70

4.7 ThesisMyCase: Users Can Specify The Folder For Log Files . 71

5.1 Comparison of the Average Admission Ratio of Handoff Users 87

5.2 Comparison of the Average Number of Admitted Handoff Users 89

5.3 Comparison of the Average Admission Ratio of New Users . . 91

5.4 Comparison of the Average Bandwidth Utilization 93

5.5 Comparison of the Average Number of All Admitted Users . 96

5.6 Comparison of Average Admission Ratio of Handoff Users with Different Ratio of Two Types of Users 100
5.7 Comparison of Average Admission Ratio of New Users with Different Ratio of Two Types of Users 101
5.8 Comparison of Average Bandwidth Utilization with Different Ratio of Two Types of Users 102
5.9 Comparison of Average Number of All Admitted Users with Different Ratio of Two Types of Users 103
5.10 Average Admission Ratio of Handoff Users Versus Average Admission Ratio of New Users 104

List of Tables

3.1 A Typical Example of A User Profile 19

3.2 Establishing The User Profile (Step 1) 23

3.3 The Table Mapping Applications And RT/NRT 24

3.4 Establishing The User Profile (Step 2) 24

3.5 Establishing The User Profile (Step 3) 25

3.6 A Sorted Degradation List Based on Bandwidth Consumed by NRT Connections 43

3.7 A Sorted Degradation List Based on Bandwidth Consumed by RT Connections 44

List of Symbols

Symbol	Definition
Δ	An insurance factor given by the equipment vendors
B_{NRT}	The bandwidth required by a non-real time application
B_{RT}	The bandwidth required by a real time application
B_{aa}	The actual available bandwidth in the BS
B_{as}	The currently used bandwidth for an application
B_b	The bandwidth upper bound subject to every class
B_c	All dedicated bandwidth the target cell holds
B_i	The initial bandwidth subject to every class
B_{il}	The initial bandwidth used by a single user whose class is the lowest class
B_{iu}	The initial bandwidth required by a user
B_r	The total reserved bandwidth to ensure the initial bandwidth for all existing users
B_u	The largest actually used bandwidth for every class
B_{ua}	The summation of bandwidth consumed by all existing users
B_w	The bandwidth (in bps) of the wire between BS and its MSC
d	The distance between the source BS and the user
N	The number of all existing users
R_{AH}	The admission ratio of handoff users
R_{AHnum}	The number of admitted handoff users

R_{AN}	The admission ratio of new users
R_{ia}	The interarrival time
S	The strength of signal received by the source BS
$Size$	The size (in bits) of the *User Profile* of a handoff user
S_{min}	The minimal value of the received signal before the BS loses the connection
T_s	The service time
t	The time
U_{BW}	The bandwidth utilization

List of Acronyms

Acronym	Definition
3G	The Third Generation Wireless Networks
3GPP2	The Third Generation Partnership Project 2
API	Application Programming Interfaces
BUB	Bandwidth Upper Bound
CDMA	Code Division Multiple Access
BS	Base Station
CLR	Common Language Runtime
DiffServ	Resources Reservation Protocol
DLL	Dynamic Link Library
DSCP	DiffServ Code Point
ECMA	European Association for Standardizing Information and Communication Systems
FTP	File Transfer Protocol
GC	Garbage Collector
GPRS	General Packet Radio Service
GSM	Global System for Mobile
IDE	Integrated Development Environment
ITU	International Telecommunication Union
MS	Mobile Station, i.e. Mobile/Cellular Phone
MSC	Mobile Switching Center

NRT	Non-Real Time
OOP	Object-Oriented Programming
QoS	Quality of Services
PSTN	Public Switched Telephone Network
RSVP	ReSerVation Protocol
RT	Real Time
TDMA	Time Division Multiple Access
TD-SCDMA	Time Division Synchronous Code Division Multiple Access
TOS	Type of Service
TPT	Transmission Power Threshold
UMTS	Universal Mobile Telecommunications System
WCDMA	Wideband CDMA
XML	Extensible Markup Language

Chapter 1

Introduction

The Third Generation Wireless Networks (3G) provides the next generation mobile service which offers better quality voice and high-speed Internet and multimedia services anywhere and anytime. The new data services require high data throughput and Quality of Service (QoS) management resulting in a demand for effective resource allocation schemes in wireless networks.

This chapter is the introduction to required knowledge employed in this thesis. We first introduce the evolution from Code Division Multiple Access (CDMA) to 3G[1]. There are two types of services in 3G in terms of the kind of data: the voice service and the data service. All problems studies by this thesis are in the area of the data service. Thus, the architecture and some features of data services in 3G will be introduced. Since the handoff problem and the bandwidth allocation problem are two problems we study, the background of these two issues will be covered afterwards. The final section is the contribution of this thesis.

1.1 From CDMA to 3G

The tide of 3G is coming. Its goal is to deliver the voice service and the data service at anytime, anywhere, any access way for mobile users. As the next generation wireless

[1] All information pertaining to CDMA and CDMA2000 in this chapter is from [1] unless specified otherwise.

networks, it evolves from CDMA but it has improved quality and enhanced data rates.

1.1.1 CDMA

CDMA was invented as the replacement of the air interface multiple access scheme. The first cellular networks, using analog radio transmission technologies, was introduced in early 1980s. Due to the high demand and surprising revenue from subscribers, cellular networks were widely deployed across the world. This rapidly brought the problem of scarce radio spectrum, resulting in dropped calls and busy signals. To solve this problem, the digital wireless technologies, including Time Division Multiple Access (TDMA) and Global System for Mobile (GSM), were created as the replacement. Although the time-sharing mechanism used by TDMA and GSM is able to improve the capacity by three or four times over the analog systems, the pressure of rapidly increasing traffic forced people to look for better solutions. In 1995, CDMA was officially launched to the market by the pioneer QUALCOMM Inc. as the newer replacement for existing wireless solutions due to its incredible ability - improved capacity around 10 times more than analog systems [1]. In addition, CDMA also brings better traffic quality and higher traffic security.

Different from the TDMA/GSM where multiple users share the same frequency in different scheduled time units (i.e. a time-sharing scheme), CDMA enables multiple users to communicate via the same frequency at the same time. In terms of capacity of spectrum, this scheme works as if it *spreads* spectrum. That is why CDMA is called a *spread spectrum* technology. This powerful capability stems from *Code Division*, which means unique codes are assigned to each pair of the sender and the receiver in the same spectrum so that the traffic of every user can be identified uniquely. To ensure these codes are unique, they are orthogonal in practice. Before sending out the signal, the sender will compute the dot product between its signal and the assigned code. When receiving the signal, the receiver will compute the dot product between the received signal and the assigned code as well. Because the codes are orthogonal with each other, other receivers only get zero after the dot

product, i.e. other users cannot correctly decode the signal though they receive the signal. This guarantees that only the expected receiver can *understand* the sender's signal though all other users can *receive* the signal. By using this mechanism, CDMA maximizes the usage of limited spectrum.

Here is an example to show the difference between TDMA/GSM and CDMA. Suppose several people are communicating with each other. In TDMA/GSM every speaker is scheduled to speak several words at particular time units and then the listener collects the heard words together to integrate them into one sentence. CDMA works in the way that every speaker speaks a different language, such as Chinese, English and etc., and the listener only understands the language used by the speaker (the unique language is equivalent to the unique code and only the pair of the sender and the receiver can understand the code since all codes are orthogonal). In this case, even if multiple speakers speak at the same time, the listener who can speak the same language is able to easily recognize the words of the speaker as long as not too many people are speaking together.

Although CDMA greatly improved the capacity of spectrum and brought better quality of to voice services, in a few years subscribers had more requests, especially for the data services. This directly resulted in the advent of 3G.

1.1.2 3G

As the next generation wireless technology evolved from CDMA, 3G promised to deliver high-speed data services and other new features. Although there was disagreement among telecommunication tycoons, the International Telecommunication Union (ITU) compromised an industry standard for 3G wireless systems.

Today, big telecommunication carriers are still arguing over the industrial standard. Three available standards are as follows and each one claims that it can beat others in some aspects. However, a common feature is that all of them are based on CDMA.

1. American proposal CDMA2000 discussed in [1]. Pioneered by QUALCOMM, the

first commercial CDMA2000 network was established in South Korea.

2. European proposal Wideband CDMA (WCDMA) introduced in [2] . In 1996, the Universal Mobile Telecommunications System (UMTS) Forum was established in Switzerland and they proposed European 3G standard WCDMA , also known as the UMTS. In 2001, the biggest Japanese carrier NTT DoCoMo declared the first commercial WCDMA networks, whose products were provided by Lucent Technologies, Ericsson and NEC.

3. Chinese proposal Time Division Synchronous Code Division Multiple Access (TD-SCDMA) introduced in [3]. In 2000, Da Tang Telecommunication Corp. released TD-SCDMA standard, which was accepted by the Third Generation Partnership Project 2 (3GPP2) in 2001. In this year (2003), Da Tang did several successful practical experiments and its commercial products will be released in 2004. Due to the huge market in China and low price of Chinese products, TD-SCDMA will have an attractive future at least in China.

This thesis is based on the CDMA2000 standard due to its existing market share. In CDMA2000, two branches should be introduced briefly.

1. CDMA2000 1X. This branch is for voice and data delivery over the standard CDMA channel. It has up to two times capacity more than TDMA/GSM and accommodates the new demand of the data services. Moreover, the peak data rate is up to 0.153 Mbps. Due to its back-compatibility, it is easy to be deployed.

2. CDMA2000 1xEV-DO. Our thesis in based on this technology and in our thesis now, *3G* will refer to CDMA2000 1xEV-DO. It was especially optimized for high-speed data delivery whose peak data rates can reach up to 2 Mbps. This speed makes it possible for some common Internet operations, such as downloading large files or watching online video clips. In addition, [1] claims that it is proved to have fairly low cost per megabyte, which is an important factor to subscribers.

1.1.3 Architecture

In the 3G vision, data services means that the mobile user is allowed to access the Internet via the cellular phone. By giving new functionalities to existing components of CDMA, 3G achieves the goal to deliver Internet data with backward compatibility. Figure 1.1 simply illustrates the relationship among CDMA components: mobile station (MS), base station (BS), and mobile switching center (MSC). In this context, MS is the cellular phone. BS is the fixed equipment for radio telecommunications with MSs [4]. It is located either at the center or on the edge of a cell and manages the MSs in its coverage area. The MSC switches the traffic sent from an MS or sent to an MS. It can connect to other public networks like the Public Switched Telephone Network (PSTN), or other MSCs in either the same network or different networks. It provides the interface for user traffic between the wireless network and other public switched networks, or other MSCs. Because our thesis discusses the data services in 3G, our MSCs connect the public network to access the Internet. When an MS accesses the Internet, the data is sent from the MS to the BS then to MSC and finally to the wired Internet. The data received by the MS is delivered in the reverse direction. Figure 1.1 shows the relationship among MS, BS and MSC. The dashed line means the range of a cell. For simplicity, this figure only covers some roles discussed in this thesis.

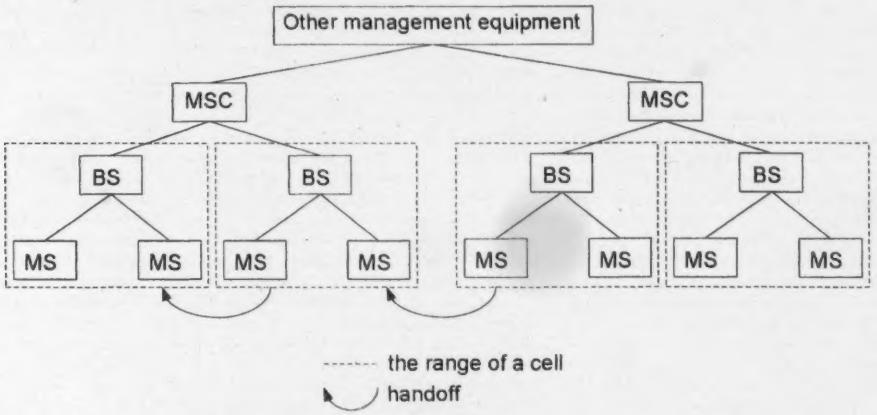

Figure 1.1. THE RELATIONSHIP AMONG MS, BS AND MSC

As we can see in this figure, each cell has one BS and one BS manages a number of

MSs in this cell. A number of BSs are managed by one MSC. Similarly, a number of MSCs are controlled by another upper management layer, which will not be covered in our thesis.

1.2 Handoffs Mechanisms in 3G

The handoff protocol in 3G is complicated because it has to deal with users who randomly roam around but ask for high-speed data delivery without disruption. Normally, there are two types of handoff mechanisms as follows:

1. Soft handoff. It means if there is overlap among coverage areas of BSs, each MS must be within *at least* one coverage area of a BS at any time. In some cases, such as on the edge of the coverage areas, the location of an MS can be covered by more than one BS, i.e. multiple BSs can receive the signal sent out from this MS at a time. In this case, the MSC will choose one BS which receives the most powerful signal to be the target cell, and this target cell will be responsible for the connection later. In soft handoff, there is no dead zone, which is the place where the signal cannot be received by any BS, because the edge of cells is thought to overlap coverage areas. Therefore, once the connection is established, *in theory*, they will not be interrupted or dropped because of the coverage area. However, for data delivery, there may be a delay in switching traffic to the new BS. Since CDMA2000 employs soft handoff, the handoff mechanisms discussed in our thesis inherit and extend the soft handoff.

2. Hard handoff. This is similar to the above soft handoff except that an MS can be covered by at most one BS at any time. Because this approach is employed by WCDMA and not used in our thesis, we do not discuss it further.

1.3 Thesis Outline and Contributions

This thesis is organized as follows. Chapter 2 is for the previous work in this area and it is a survey of related literature. Chapter 3 is the core of this thesis. We discuss in detail our

bandwidth allocation mechanism with some new concepts. To evaluate our new approach, we developed two simulation systems. One runs the conventional mechanisms and the other executes our bandwidth allocation mechanism. In Chapter 4, we explain clearly our simulation systems, including the modelling and the development. Chapter 5 presents the final simulation results and the corresponding performance evaluation. The performance analysis shows that our approach is better than the conventional approach in all metrics. The final part is Chapter 6 with conclusions and future work.

This thesis addresses the problem of how to effectively and efficiently allocate bandwidth. We develop an integrated bandwidth allocation and bandwidth degradation scheme using the concept of a *User Profile*. The final simulation experiments show that our approach is much better in all metrics. The new contributions of this thesis are as follows:

1. We propose that the carrier specifies the initial bandwidth for real time connections for every user. This initial bandwidth is based on the class users belong to, i.e. the fee users pay monthly, and the carrier ensures that (to the best of its ability) this amount of bandwidth will be available for real time connections for each user at any time.

2. In the case where bandwidth is exhausted, we propose a concrete solution of bandwidth degradation so that the BS is able to admit more users with the least impact on existing users. This algorithm is based on the realization of the difference between the bandwidth for real time connections and non-real time connections. This bandwidth degradation algorithm makes efficient use of the resources and improves best effort admission.

Chapter 2

Literature Review

Compared with wired networks, wireless networks have many intrinsic challenges. The most different feature is the mobility of the mobile host, which makes it difficult to predict the movement of the mobile host. As a result, the BS might not have *enough time* to reserve *enough bandwidth* to accommodate handoff users. Therefore, handoff users in wireless environment easily experience degraded QoS. Moreover, wireless networks are always bandwidth constrained compared with wired networks. When Gigabit Ethernet [5] was already the fact in wired networks, CDMA2000 just reached a maximum throughput of 2 Mbps under special condition. This great difference lets Quality of Service (QoS) more difficult to be achieved or guaranteed. Another intrinsic weak point of wireless networks is that wireless signals always experience interference, high-loss rate and high data error. Due to this feature, the quality of signal and quality of transmission are more difficult to be guaranteed and are greatly time-varying.

This thesis essentially studies QoS issues in wireless networks. The answer to the question of providing QoS in cellular networks is a combination of admission control and bandwidth allocation, though many papers only discuss one of them instead of both. There are many papers discussing how to allocate bandwidth and QoS issues in handoffs. A brief description of some of the literature is given here.

2.1 Bandwidth Allocation Mechanisms Dealing with Handoffs

2.1.1 DiffServ and Its Variants

Provision of QoS is discussed at the network level for the Differentiated Service Protocol (DiffServ) in [6], [7] and [8]. The general idea of DiffServ is to employ the Type of Service (TOS) field in the IP header to classify the traffic and set up priorities. Different classes are given different QoS treatment depending on the service level agreement between the carrier and the subscriber. In practice, the TOS field is set up when the packet is transmitted through the network boundaries (Autonomous System boundaries, internal administrative boundaries, or host boundaries). Then, based on the requirements or rules of each service, the packets with different TOS are treated differently at network boundaries. DiffServ can work well in wired environments. However, [6], [7] and [8] do not study how to deploy DiffServ in a wireless environment, which has high loss rate and low bandwidth constraints.

[9] realized the above problem and discussed how to extend DiffServ to wireless networks. First, the traffic *classifier* and *conditioner* should be modified by using a modified Class Based Queuing (CBQ) introduced in [10]. The classification of traffic is still based on TOS, but the packet dropping scheme is extended and the bandwidth allocation scheme for a class is modified as well. Moreover, the signalling protocol is designed and the signalling information is delivered via ICMP messages. A *Report* message is sent from mobile host to BS to transmit the mobile parameters, such as bandwidth requirements. Then, a *Reply* message is sent from the BS to mobile host to deliver ACKs. Finally, a *Modify* message is delivered from mobile host to BS again to indicate the bandwidth requirement for a particular session if there is any change of the requirement of this session. Although this paper proposed other required modifications to extend DiffServ to wireless networks, we think the most attractive idea is *degradation*, which means that the currently admitted users will have their previously assigned bandwidth taken away to deal with incoming handoff users. However, this paper did not clearly describe the degradation mechanism. Based on this

idea, after we distinguish the difference between real time connections and non-real time connections, our thesis proposes the entire degradation algorithm.

2.1.2 Guarded Channel Scheme and Its Variants

The trunk reservation scheme (i.e. the guarded channel scheme) introduced in [11] and [12] is widely used in traditional voice centric cellular networks. This idea is simple and practical: the BS reserves an amount of bandwidth especially for handoff users and the rest of the bandwidth is used for other users. When the rest of the bandwidth is exhausted, the BS begins to use reserved bandwidth to admit handoff users. If the reserved bandwidth is exhausted, the BS starts to reject handoff users.

[13], [14] and [15] extend the above guarded channel scheme by queuing the requests from handoff users and having the BS minimize the rejection by blocking new calls. This means that the BS has a waiting queue for requests from handoff users and the size of this queue is predefined. If the resources are exhausted, all requests of this kind are enqueued to wait for the admission till there is enough available bandwidth. During such a waiting time, more requests from handoff users will either be enqueued (if the queue is not full) or directly rejected (if the queue is full). Also based on the guarded channel scheme, [16] proposed a fractional guarded channel scheme to achieve an optimal solution. However, all these guarded channel scheme variants cannot be timely adaptive to network changes. This makes these variants not appropriate in wireless networks where there are frequently bursty data. In that case, a more dynamic and adaptive scheme is needed.

[17] proposed the predictive bandwidth allocation. The idea is that once the mobile user sends the connection request, it also supplies the bandwidth requirement and some accurate information about the mobility, such as the current position and the movement. This information will be sent to both the BS in the current cell and the BSs in the adjacent cells, then all these BSs reserve the bandwidth in advance on the basis of the mobile host's requirements. Apparently, this scheme ensures that the mobile host will be guaranteed QoS

no matter which cell it roams to. However, this mechanism lets the adjacent cells waste the bandwidth because all adjacent, normally 6 cells, have to reserve bandwidth though only one of them will finally use such reserved bandwidth. In busy areas where a large number of mobile users frequently handoff from one cell to another, by using this scheme, the adjacent cells might often miss the chance to admit new users entering themselves because their bandwidth is quite often reserved for users currently moving into their adjacent cells.

[18] emphasized that once the BS receives a connection request, the corresponding bandwidth is allocated for the current cell, where this user is located, and all adjacent cells. After the mobile host is admitted by a destination cell, all other cells release the bandwidth previously reserved for this user. The feature of this approach is that the amount of reserved resources can be dynamically adjusted by monitoring the average connection dropping probability and the reserved bandwidth utilization. This scheme differentiated the new users and handoff users by assigning higher priority to the latter. Like the above paper [17], this approach also wastes the bandwidth more or less since 6 cells reserve the bandwidth but eventually only one cell will make use of such reserved bandwidth.

An adaptive and dynamic bandwidth allocation algorithm was studied in [19]. The first feature of this approach is that the required bandwidth is estimated online and periodically. This guarantees the change of bandwidth during the transmission duration will be timely and accurately monitored. It is very suitable to the wireless networks, where the data rate is changed quite often. The other feature of this scheme is that the bandwidth allocation is done by a probabilistic mechanism, which means that the BS allocates the bandwidth to a specific user in a statistic multiplexing manner. The authors give formulas for that probabilistic mechanism and claimed that these formulas are accurate enough so that it is unnecessary to reserve the bandwidth anymore.

[20] for the first time proposes the notion of *utility function* for network resource management. However, this notion only presents a generic model, i.e. it only analyzes how the adaptive approaches (including a strongly adaptive approach and a weakly adaptive ap-

proach) influence the utilization and does not discuss the specific scheme. On the basis of [20], [21] studies a utility-based approach which has quantitative adaptation instead of the qualitative one in [20]. [21] proposes a split level adaptation control, which is actually a complex framework. This adaptation control consists of the network level and the application level. The former will periodically probe the wireless links and then the bandwidth reservation is adaptive via messages of signalling protocol for reservation like RSVP [22]. The latter relies on a number of distributed adaptation handlers, which are components in the framework, to achieve the application-specific adaptation. Apparently, the entire solution is complicated especially the framework. Besides, the predefined periodical interval directly influences the accuracy of reporting the bandwidth reservation situation. In wireless networks where the traffic is greatly time-varying, it is difficult to use a constant periodical interval to report the bandwidth reservation situation.

[23] also proposes a degradation mechanism. In its environment of GSM/General Packet Radio Service (GPRS), a user might be assigned multiple channels[1] and in the future this user can have extra channels taken away in the degradation process to admit handoff users. Authors realize that too much degradation of admitted users might result in poor QoS because traffic studied in this paper are real time data. This literature defines a parameter of *degradation tolerance* to indicate how much degradation a connection can tolerate and the constraints, previously negotiated at the time of establishing the connection, are still guaranteed. It is good to realize that a real time connection cannot tolerate the degradation without limitation. However, rather than choosing degradation based on the requirement asked by the mobile host at the time of connection setup, we use an *initial* guaranteed bandwidth promised by the carrier based on the monthly fee paid by the subscriber (more detailed information can be found in Chapter 3).

[1]GSM/GPRS is the hybrid of time division and frequency division mechanism. This means that the BS assigns frequencies to a number of groups of mobile users, i.e. frequency division. To each frequency, a group of mobile users share this frequency in the time-sharing manner, i.e. time division scheme. Due to different bandwidth requirement of different users, a user might be able to use multiple time slots, i.e. multiple channels in this context.

In addition, priority is considered in this paper. Some channels are given higher priorities and connection requests with higher priority are given channels with higher priority. They set up the higher priority for real time connections over non-real time connections and non-real time calls will be put in a priority queue. In our thesis, we differentiate real time traffic and non-real time traffic and consider the priority as well. However, we investigate the different treatment for traffic with different priorities by assigning different amounts of resources.

The scheme in [24] proposes that every used channel can be split into two equal-sized channels if the bandwidth is exhausted. One channel is still used by the owner (the user whose channel is split) and the other channel is used for the handoff user. Apparently, fixed size sub-channel will waste the bandwidth if the bandwidth request does not exactly match the fixed size. And this situation might be likely occur in practice. In addition, it is hard to convince that the left half channel can guarantee QoS of the admitted user.

2.1.3 RSVP and Its Variants

The ReSerVation Protocol (RSVP) originally proposed in [22] makes resource reservation for users in wired networks via end-to-end signalling. [25] considers the use of modified version of RSVP in wireless networks. This literature assumes that all BSs and routers understand passive reservation [26] [27], i.e. the bandwidth is reserved along the transmission path but the data is not being delivered long this path. The modification of RSVP should integrate the passive reservation in RSVP and CBQ extensively discussed in [28]. For example, the PATH message of RSVP should include the information to indicate whether or not the reservation is passive. In addition, the modified version should integrate QoS parameters which are specific to the mobile environment, such as the probability of seamless communication [29]. Finally, people should modify RSVP so that it is able to work with the passive reservation mechanism. Because the modification is major, the revised version of RSVP is complicated to implemented.

2.1.4 Admission Control

In [28] admission control for integrated voice and data traffic in packet radio networks is investigated. The admission control policy is based on a pre-determined threshold value of either the mean delay or the packet loss probability for data traffic and on the long-term blocking probability for voice traffic. This scheme does not reserve bandwidth in neighboring cells and does not adapt to changes in network conditions.

In the admission control scheme proposed in [29], different resource sharing schemes are employed to allocate resources to different classes of traffic. Although this scheme is interesting and useful, it does not consider reserving bandwidth in advance. It only assumes a simple Poisson traffic model. A distributed admission control scheme is proposed in [30]. In this scheme, the admission control is based on both the number of existing connections in the cell where a connection request is generated and the number of connections in the adjacent cells. Admission control is performed at each base station in a distributed manner. Only a single traffic type is considered. Unlike our proposed scheme, this scheme does not reserve bandwidth and is not adaptive to changes in network conditions. Therefore, the scheme proposed in [30] may not work well if the network carries diverse types of traffic with varying bandwidth requirements since the number of connections alone may not provide accurate information regarding the network traffic load. In [31], an admission control based on dynamic channel assignment is proposed. In this scheme, network traffic conditions are first evaluated in the admission control phase, and then channels are assigned accordingly to new calls. Although this scheme is adaptive to changes in network conditions, it does not reserve bandwidth in neighboring cells. Therefore, this scheme risks an inability of meeting QOS requirements when hand-off occurs.

2.2 Summary

In this chapter, we investigated a number of related researches in the literature pertaining to QoS issues in wireless networks. Researchers proposed various approaches to solve this

problem.

Because DiffServ and RSVP work well in wired networks, people have thought to extend them into wireless networks. However, DiffServ requires all nodes along the transmission path to understand this mechanism and its complexity cannot be ignored. Although its bandwidth allocation scheme for a class was modified, it only distinguished the difference between classes and did not explicitly differentiated the great difference between real time and non-real time traffic. Moreover, since all message exchanges are transmit through ICMP, the non-reliable transport scheme, it cannot ensure that important information about QoS can be receive by the receivers. In this case, it is much harder to guarantee QoS. Since RSVP requires end-to-end signalling, it is harder for RSVP and its variants to timely respond to changes in the wireless links, whose characteristics are greatly time-varying.

People have studied the guarded channel scheme, a successful scheme in voice communications. Its essential idea is that the BS should reserve some resources to deal with handoff users. People proposed various mechanisms about how much bandwidth should be reserved and in which manner the BS should handle users. They have guaranteed some bandwidth, however, deciding the amount that should be guaranteed is a difficult problem.

This thesis extends the previous work and tries to solve two problem by combining admission control and bandwidth allocation. We propose a mechanism which can be used by the BS to know the QoS requirement of this incoming user in advance. The other problem is how to allocate bandwidth to users, especially in case when the bandwidth appears to be exhausted. Based on some previous papers, we explicitly differentiate the classes, real time traffics and non-real time traffics, and new users and handoff users. This means we give different priorities depending on classes, the types of traffics, and the type of users. Moreover, this thesis proposes a complete degradation algorithm and the simulation shows that it works well.

Chapter 3

The Bandwidth Allocation Mechanism

In this chapter, we present our novel mechanism to efficiently allocate the bandwidth to users in a cellular network. The feature of this mechanism is that it is dynamic, application-aware and class-based with the least impact on the admitted users.

The bandwidth allocation solution in this context is the bandwidth allocation algorithm for both handoff users and new users (those location in the cell and asking for connection admission for the first time) using data services in 3G environments. We propose a new algorithm to effectively and smoothly handle the handoff users and provide a degree of QoS to handoff users.

At a high level, we enable our bandwidth allocation algorithm to be class-based and application-aware. Class-based denotes that the different classes of users are recognized and they are assigned different priority when the BS allocates the bandwidth. In addition, the class also indicates the different initial bandwidth which is the smallest bandwidth a user will be assigned to RT connections once admitted. Therefore, the different users are treated differently depending on their classes, which reflects the service they have paid for.

The mechanism is also application-aware in terms of the time sensitivity of connections. Different connections are categorized into real time and non-real time connections launched by corresponding applications. We believe that a real time connection should be assigned enough bandwidth and keep those assigned resources during the entire transmission time. On the other hand, the non-real time connections have fluctuating requirements for the

bandwidth and higher time delay tolerance. Realizing this feature is very important because in the case of exhausted resources, it allows the BS to *temporarily* take away the assigned bandwidth for the non-real time connections for a certain period of time and then assign the bandwidth back to the non-real time connections. In the meantime, the BS does not touch the resources assigned to the real time connections. The idea is that we treat connections differently depending on the characteristics of the connections - the time sensitivity. Thus, by using this feature, the BS could admit more users (including handoff users and new users) with small impact on the non-real time connections and *zero* impact on the real time connections, i.e. the user experience is maintained as much as possible.

In order to accurately record the useful characteristics of the connections, we propose to use a *User Profile* table. The *User Profile* is built tuple by tuple by the BS during the process when the mobile user launches some applications. Because the *User Profile* records the QoS requirements for all connections for one user, this table could be forwarded between BSs in case of handoffs so that the target BS can know the mobile users' requirements in the source cell and then the best effort is done to timely handle handoff users and allocate bandwidth based on their previous requirements. Since the *User Profile* is forwarded between BSs, we carefully select the metrics for characteristics of the connections to minimize the size of the table.

The proposed degradation algorithm is triggered only when the resources on the BS side are exhausted. Conventionally, the BS indifferently rejects all admission requests in this case. But considering the difference between the real time connections and non-real time connections, we propose to temporarily degrade the bandwidth consumed by the non-real time connections to save the bandwidth to admit the admission requests. Due to the tolerance of time delay of non-real time connections, temporarily taking away resources may not greatly influence these connections.

This chapter is organized as follows. The first section explains the proposed table *User Profile* describing the QoS metrics of each connection. The next section is the critical part

explaining our bandwidth allocation algorithm in detail. In the final section, we show the entire working process of the source BS from the moment when the MS normally works with the source BS to the moment when the source BS finally terminates the connection with the MS, who has already left.

3.1 User Profile

In order to solve QoS-related questions, at first we have to find some appropriate metrics to evaluate various aspects of QoS. In our novel mechanism, we store information describing the important characteristics of the connection for all currently admitted users in a cell. We name this information the *User Profile*. This is stored in the memory of the BS and updated once any user starts (or closes) one connection or a daemon process realizes the consumed bandwidth is changed. For each mobile user, the *User Profile* records the class the mobile user belongs to, the types of currently running applications, and the bandwidth it consumes.

In this section, we first introduce the proposed *User Profile* in detail. Then we will explain how the BS establishes and uses the *User Profile* for a roaming user. This is followed by the summary.

3.1.1 The Structure of The User Profile

The *User Profile* is actually a table recording QoS-related metrics for every connection launched by the user. The requirements for the *User Profile* are that the table must be *accurate* so that these carefully chosen metrics are able to clearly describe the current status of the QoS situation; and it also must be *concise* so that the transmission of this data will cost the least time. Table 3.1 is a sample of a typical *User Profile*.

As we can see, the *User Profile* records user ID, the type of class to which the mobile user belongs, the currently running applications and their corresponding consumed bandwidth. We should notice that the applications are categorized into two types: the real time applications and the non-real time applications; and the last column respectively records

Table 3.1. A TYPICAL EXAMPLE OF A USER PROFILE

ID	Class	Initial B/W (Mbps)	Running Applications	
			Type of Applications	Consumed Bandwidth (Mbps)
User 1	B	0.384	Real Time	0.4
			Non-Real Time	0.1
User 2	A	2.0	Real Time	1.5
User 3	C	0.144	Non-Real Time	0.3
			Non-Real Time	0.2
			Real Time	0.2
User 4	A	2.0	Real Time	0.7
			Non-Real Time	0.3
			Non-Real Time	0.5

the consumed resources for the different types of applications.

User ID

The first column in Table 3.1 is *User ID*. This ID number is used to uniquely identify the mobile user. If the user is the handoff user, the information of *User ID* will be forwarded from the source cell to the target cell via the Mobile Switch Center (MSC) (more detailed information will be found in 3.1.3). Thus, the target cell is able to identify the user based on the forwarded information. If the user is a new user, the process of identification is slightly more complex. In the management architecture for mobile users, there is a database which stores user IDs and their corresponding information, including whether or not this user is an authorized user registered in a carrier, which service this user is authorized to use and how much is the initial bandwidth this user pays for. Once the BS identifies a user as an unknown user, it forwards report to its upper layer the MSC, which either identifies the user is a handoff user roaming from another cell or a new user after querying information from that database.

The Class of Service

The second column in Table 3.1 is *the class of service*. Normally, the carrier provides several kinds of services, named *the class of service*. Every mobile user is assigned to a class based on the fee this user pays monthly. A *class* in this context also means a different initial bandwidth the user is allowed to consume. That is the carrier guarantees that the bandwidth provided by the carrier to a user is equal to or larger than the promised initial bandwidth of the class of the user. The coined term *initial bandwidth* here refers to the bandwidth initially assigned by the BS to the user's RT connections when this user is admitted.

Higher classes would have larger initial bandwidth. For example, depending on the monthly fee paid by *User 1*, the bandwidth this person can use is at least 1 Mbps when the BS assigns resources to *User 1* initially; while another user, *User 2*, in a higher class may be guaranteed at least 1.5 Mbps. Thus, the class of service is an important criterion to evaluate the situation of bandwidth consumption.

Another important use of the information of class is that users in a higher class are treated with higher priority than users in a lower class when in competition for bandwidth. A typical example is that in an environment with limited resources, two users with different classes might request an amount of bandwidth simultaneously. If the available resources only can admit one user, the user in the higher class will be served and the other user in the lower class will be rejected.

Therefore, the class of service indicates different initial bandwidth listed in the third column, different priority and accordingly different treatment. The class is an important criterion for deciding how to treat the user especially in the case of handoffs in a resource-limited cell. This will be discussed further with the admission requests, bandwidth allocation and bandwidth degradation policies.

The following rule reveals the relation between the class of service and how a user would be treated.

Rule 3.1 *The class of service indicates the priority of the service, which is used for class-based treatment.*

Running Applications

The fourth and the fifth columns in Table 3.1 contain information for *the applications running by the user*. The running applications are classified as *real time applications* and *non-real time applications*, which indicates the sensitivity of the delay. In this context, *real time* (RT) denotes that this application is delay sensitive and requires real time data delivery. In contrast, *non-real time* (NRT) means that the application is not as concerned about the delay. Precisely, the level of sensitivity is subject to different applications on different systems. For example, a voice connection over the traditional telephone switching system can only tolerate delay in the order of a millisecond but the web browser can tolerate delay in the order of a second. However in this thesis, we only study the common application used by the mobile users to access the Internet, such as the browser (as a non-real time application) checking email, or the FTP software (as a non-real time application) downloading some binary codes, or the Windows Media Player [1] (as a real time application) playing a video clip. The corresponding consumed bandwidth is recorded in the fifth column.

As we explained before, the class of service indicates the initial bandwidth guaranteed by the carrier once the user asks for the bandwidth. However, how much bandwidth a user *actually* consumes will normally be different from the initial bandwidth. The user might only need a small amount of bandwidth, like launching an application to check email, in which case the required bandwidth is only 100 Kbps. Another possible case is that a user might ask for more bandwidth than the initial bandwidth when there *exists* a lot of available resources. In this particular scenario, not meeting the user's request obviously wastes the resources and decreases the user experience. If a BS assigns more bandwidth to this user, the consumed bandwidth is accordingly more than the initial bandwidth. Hence, the actually consumed bandwidth is not subject to the initial bandwidth, which is derived

[1] Windows Media Player is the trademark registered by Microsoft Corp.

from the class of users. Instead, it should be subject to the currently running applications. In our design, a daemon will periodically record the current consumed bandwidth for every existing connection and write this information into the corresponding tuple. The concrete process can be found in Section 3.1.2 *Establishing the User Profile*.

The reason why we pay attention to the sensitivity of the connection, i.e. whether or not it is RT or NRT, is that we can take advantage of their characteristics. Because NRT applications can tolerate longer delay and the connection does not continuously require data delivery, dedicating an amount of bandwidth to this application will waste resources during the time when the connection does not need data delivery at all. For example, suppose checking email requires 50 Kbps to do its work. Due to the characteristic of this software and the behavior of the user, the bandwidth consumed by this email checker varies from time to time: it consumes 50 Kbps when it downloads a new email to read or it sends out a reply; and it consumes *zero* resources when the user is reading an email or writing the reply. If we still hold the bandwidth to 50 Kbps to the user in the latter case, we obviously waste the resources because no one else is allowed to use such bandwidth at this moment. Furthermore, the user can tolerate some delay. In contrast to NRT, the RT application strictly needs the dedicated resources, i.e. it continuously requires an amount of bandwidth during the entire transmission period.

By recording the information pertaining to applications, the system has some way to know the bandwidth consumption in the client side at the level of the application, instead of solely knowing how much a user consumes. Thus, the granularity is greatly improved.

In consideration of the utilization of resources, we believe that the system should treat RT and NRT applications differently based on the following rule due to the above different features of connections.

Rule 3.2 *To RT applications, the BS should reserve an amount of resources during the entire transmission period once it assigns some resources to the user. To NRT applications, the BS should assign some resources to the user only when the user requests, and later the*

assigned resources could be taken back by the BS to assign to another user because of the delay tolerance. We name this approach the application-aware mechanism.

3.1.2 Establishing the User Profile

The *User Profile* is created by a BS and stored in the memory of the BS. Every BS has a *User Profile* which only records the information for the users currently controlled by this BS. If a mobile user is managed by a BS, the corresponding tuple in the *User Profile* is stored in this BS. Therefore, any mobile user only can appear in one *User Profile* because any user only can be managed by one BS at any time. The *User Profile* entries are established step by step as follows.

To avoid ambiguity, in the *User Profile*, the information associated with one mobile user is called *a tuple*, and the information associated with one connection of one mobile user is called *a row*. For example, in Table 3.5, all information pertaining to the *User 1*, including the above two rows, is regarded as *a tuple*. The information for "Real Time" and "0.3 M" is regarded as *a row*, and the information for "Non-Real Time" and "0.1 M" is also regarded as *a row*. Thus, one tuple of the *User Profile* corresponds to a unique user and one row corresponds to a connection.

When a new user is admitted by a BS, a new tuple is added into this profile. This tuple includes the user ID and its corresponding class since this information is retrieved from the admission request. At this moment, there is no information for *Running Applications* because the connection is not yet established. Table 3.2 gives an example of the *User Profile* at this time.

Table 3.2. ESTABLISHING THE USER PROFILE (STEP 1)

ID	Class	Initial B/W (Mbps)	Running Applications	
			Type of Applications	Consumed Bandwidth (Mbps)
User 1	B	0.384		

When an admitted user launches an application to access the Internet, the user sends the request to the BS and this request tells the BS what the application is and how many resources it needs. Since the mobile user has a limited number of applications to access the Internet, such as a browser or an online MP3 player or an email checker, the BS can keep a small table mapping the relation between every application and the characteristics of the connection it requires, i.e. RT or NRT. Table 3.3 [2] shows a typical example of a mapping table.

Table 3.3. THE TABLE MAPPING APPLICATIONS AND RT/NRT

Application	RT/NRT
Windows Media Player	RT
Outlook	NRT
RealOne Player	RT
Acrobat Reader	RT
Internet Explorer	NRT

With the help of this pre-stored table, the BS easily understands the metric of RT/NRT once the user sends the request to the BS. If the BS allows this request, more related information, including the type of application (RT or NRT) and their corresponding consumed bandwidth asked by the user, is stored into the tuple of this user. Thus, an entire tuple is completed, which is showed in Table 3.4.

Table 3.4. ESTABLISHING THE USER PROFILE (STEP 2)

ID	Class	Initial B/W (Mbps)	Running Applications	
			Type of Applications	Consumed Bandwidth (Mbps)
User 1	B	0.384	RT	0.4

In the real world, a user is allowed to launch multiple connections to access different links (or web sites). Thus, if the same user starts more than one applications to establish

[2]Outlook, and Internet Explorer are trademarks registered by Microsoft Corp.. RealOne Player is the trademark registered by RealNetworks Inc.. Acrobat Reader is the trademark registered by Adobe System Inc..

the Internet connections, multiple rows for the same user ID will be stored in the profile by the approach discussed above. The BS is not concerned with the number of RT connections or NRT connections providing there is enough bandwidth. Instead, the BS solely monitors the total consumed bandwidth, which should be lower than a certain degree (this degree is associated with BUB discussed in detail in 3.2.1). The typical example in this case is shown in Table 3.5.

Table 3.5. ESTABLISHING THE USER PROFILE (STEP 3)

ID	Class	Initial B/W (Mbps)	Running Applications	
			Type of Applications	Consumed Bandwidth (Mbps)
User 1	B	0.384	Real Time	0.4
			Non-Real Time	0.1

The *User Profile* will be dynamically updated to reflect the change of the actual consumed bandwidth for each connection. A daemon process in the BS is responsible for this update. To a real time application, the BS will do its best to guarantee the consumed bandwidth even though handoff normally, and guarantee the initial bandwidth in the bandwidth degradation process (the bandwidth degradation process will be discussed in detail in Section 3.2.2). When the connection is established, the daemon only writes the record into the *User Profile*, including the real time application and its corresponding consumed bandwidth. The daemon removes this tuple when the connection closes. To the non-real time application, once this connection has some data being transmitted through the BS, the daemon will record the currently consumed bandwidth for this connection. When a non-real time connection has no data transmitted through the BS, the value of *Consumed Bandwidth* of this connection is set to be zero. Therefore, by dynamically updating the column *Consumed Bandwidth*, the daemon ensures that the column in the *User Profile* reflects the up-to-date situation of the consumed bandwidth of the connections.

Because our novel mechanism sometimes allows the users to use more bandwidth[3] than the initial bandwidth (more detailed information can be found in Rule 3.5), the daemon will update the corresponding *Consumed Bandwidth* once the BS assigns more resources to a connection.

Releasing the connection or closing the application is done using the reverse process. While a user closes an application to release a connection, the user still sends the release request to the BS and this request tells the BS which connection will be closed. By checking the *User Profile*, the BS closes the connection and removes the corresponding row from the profile. The tuples of the *User Profile* remain temporarily when the user roams into another cell, according to the handoff process as explained later.

3.1.3 How to Apply the User Profile to The Handoff User

The functionality of the *User Profile* is to store the critical metrics of QoS for every application running on the client side. If the user remains within a cell, the BS solely lets the daemon timely update this table and establish or remove some tuples. However, in the case of handoffs, the BS has to do more complicated things. Once a handoff occurs, the source BS forwards the corresponding tuple in the *User Profile* to its upper level management equipment the Mobile Switching Center (MSC). Then the target cell will receive the information of this incoming user from the MSC. Again, the handoff in this context is the soft handoff, introduced in Section 1.2 in Chapter 1.

While the MS is close to the boundary of cells up to a certain degree, the BSs in both the current cell and the adjacent cells can receive the signal from this user. For instance, in Figure 3.1, when the user is moving from the position *P1* in *B1*'s cell to the position *P2* in *B4*'s cell, both *BS1* and *BS4* can hear this user. In our extended soft handoff mechanism, when the *BS1* is notified from the system that the MS switches to the handoff state, the *BS1* forwards to the MSC the corresponding tuple of *User Profile* of this MS. Since the MS

[3]The bandwidth here means the total bandwidth consumed for by connections instead of the bandwidth only consumed by RT connections or NRT connections.

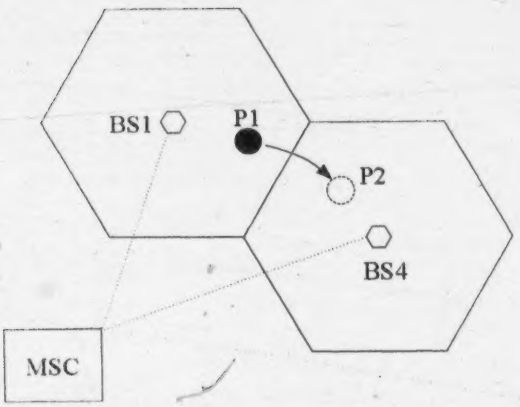

Figure 3.1. A TYPICAL EXAMPLE OF FORWARDING USER PROFILE

can receive signals from multiple BSs in handoffs, the system chooses a target BS, which has the strongest strength of the signal, to manage the MS. Then, the MSC forwards the tuple of *User Profile* to this selected target BS. The following pseudo-code illustrates this process.

```
01  // in Figure 3.1
02  //an MS is moving from P1 in BS1's cell to P2 in BS4's cell;
03  //if it is close to the boundary of cells ...
04  BS1 is notified by the system that the handoff will happen;
05  BS1 forwards to the MSC the corresponding tuple of User Profile;
06  //now, an entire User Profile table is stored in BS1 and a copy
07  //       of corresponding tuple is stored in MSC
08  meantime, this MS can receive signal from BS4;
09  MSC chooses a target cell (TC) with stronger received
10            transmission power;
11  if (TC == BS1) {
12      MSC removes that tuple from the User Profile;
13      MSC notifies BS1 to continuously handle the user;
14  }
15  if (TC == BS4) {
16      MSC forwards that tuple of User Profile to BS4;
17      MSC notifies BS4 to be the controlling BS;
18      BS4 starts the admission process based on the forwarded tuple
19                    of User Profile;
20      BS4 reports the result of admission to MSC;
21      MSC informs BS1 to close the connection with this user and
22                    release the resources held for this user;
23  }
```

If the chosen target cell is *BS4*, it means that the user has actually entered the adjacent cell and the actual handoff occurs. The MSC should immediately send the forwarded tuple of the *User Profile* to *BS4* so that it can use this information to start the admission process to decide whether or not to admit this user. If the chosen target cell is *BS1*, it means that the MS goes back to the coverage cell of *BS1* and there is no actual handoff. Thus, there is no need for the MSC to forward the tuple of the *User Profile* back to *BS1* since the entire *User Profile* table is still stored in *BS1*. The corresponding tuple of the *User Profile* stored in the MSC loses the value of its existence and should be removed in order to save the memory of the MSC.

More situations should be discussed here about the case when *BS4* is assigned to be the target BS. If *BS4* cannot admit this handoff user due to not enough bandwidth, *BS4* reports

to the MSC that this handoff user has to be rejected. Then, the MSC will notify *BS1* to close the connection with this user, release all resources held for this user previously and remove the corresponding tuple from the *User Profile*. In this case, *BS1* will not try to deliver data to the user anymore since *BS1* is not responsible for this MS anymore. If this handoff user can be admitted by *BS4*, *BS4* reports to the MSC *after* the new connection between *BS4* and the user is established. Then, the MSC notifies *BS1* to close the original connection with this MS, release all resources previously assigned by *BS1*, and remove the corresponding tuple of the *User Profile*. *Line 20-22* shows that regardless of the determination made by *BS4*, the source BS should close the connection with this handoff user because either this user is being managed by the target cell or the source BS is not supposed to be responsible for this user anymore.

The storage place of *User Profile* is different in different cases. In the case other than the handoff, the corresponding QoS information for an MS, expressed as a tuple of the *User Profile*, should be only stored in the controlling BS, which is responsible for managing the connection between the BS and this user. During the handoff, before the target BS is determined, the corresponding tuple can be stored in the MSC (temporarily) and the source BS; after the target BS is determined, such tuple is forwarded from the MSC to the target BS and the tuple in the source BS and the MSC will be removed.

The above discussion indicates how to use the *User Profile* during the handoff situation in practice, which should always follow this rule.

Rule 3.3 *The User Profile is used in this way: the controlling BS stores the User Profile in its memory; when an MS handoffs, its tuple in the User Profile is forwarded from the source BS to the target BS via the MSC; this tuple is stored in the source BS, temporarily in the MSC (during handoffs) and finally in the target BS. A row of the User Profile is removed once the corresponding connection is closed, and a tuple of the User Profile is removed once the corresponding user leaves or turns off the cellular phone.*

We must notice that the releasing process should start *after* the admission result in

the target cell is clear. This means that *line 18-19* must be executed before *line 21-22*. The target cell decides to either admit or reject the user based on the availability of its resources. If the source BS closes the connection before the new connection is established in the target cell, the packet delivery may be interrupted since there is no route of data delivery *yet* from MSC to the target cell. In this case, the data delivery from the sender stops in the MSC, which means the original connection is actually closed in the "last mile". If the handoff user is rejected by the target cell, since the source BS is notified that it is not responsible for the data delivery to the user anymore, the source cell also has to close the connection with the handoff user and then release the bandwidth consumed by this user. In any case, the source BS must wait for the admission result made by the target cell, which is indicated in the following rule.

Rule 3.4 *In handoffs, before the source BS actually closes the connection, it should wait for the admission result made by the target BS via the MSC.*

3.1.4 Summary

In this section, we introduced a new concept, the *User Profile*, a critical component in our bandwidth allocation algorithm and bandwidth degradation algorithm. Its functionality is to use the least data to accurately describe the QoS characteristics of connections at the level of the application.

The most important advantage of the *User Profile* is that it is *application-aware*. Our approach considers more details than the conventional way, including the characteristics of running applications and their corresponding consumed bandwidth. For example, to *User 3* in Table 3.1, the granularity of the traditional approach is to only understand that this user consumes 0.9 Mbps. Our approach not only contains this information, but also monitors how many applications are running by this user, the sensitivity of each application and how many resources are consumed by each application. Thus, our approach goes one more step in terms of granularity because it considers QoS issues at the level of the application instead

of the user. Knowing information like this makes it possible for the system to purposely assign resources to different applications running for different users, instead of assigning the bandwidth to an *entire* user. Thus, the system handles the bandwidth allocation in an application-oriented way instead of a user-oriented way.

The different running applications consume different amounts of resources. Thus, the consumed bandwidth is actually the summation of resources consumed by every application. This approach truly reflects the composition of consumed resources. Another advantage is that this approach makes our mechanism an application-aware mechanism, i.e. the system is aware of the information of running applications in the client side with the help of the *User Profile*, and then adjusts the assigned resources based on such information.

The metrics in this approach are *more accurate*. We not only consider the widely used metric, consumed bandwidth, but also notice a useful factor, RT or NRT. This plays an important role in our approach. In the past, the BS solely knows to allocate some resources to a specific user, but the BS does not know how the user consumes these resources and more importantly, the BS does not know the different requirements for different applications. Differentiating the concepts of RT and NRT, the BS will differently treat users at the level of the application, i.e. the BS allocates bandwidth for an *application* instead of the fuzzier notion of a *user*.

3.2 Bandwidth Allocation Mechanism

The proposed bandwidth allocation mechanism is designed to handle both new users and handoff users. It is a class-based and application aware mechanism. The best effort is done to admit the handoff users even when resources appear exhausted. New users here mean the users who start to access the Internet for the first time in any cell. When the BS is running out of bandwidth, it generally starts to reject users, no matter wether or not they are handoff users or new users. But we believe there are still many potential resources in this case once we understand the difference between the RT and NRT connections. Thus, we propose the

algorithm, named the *degradation algorithm*, to particularly handle the handoff users once the bandwidth in the target cell is exhausted.

This section has three parts as follows. The first will explain in detail how to allocate bandwidth to new users and handoff users. The second part will introduce the proposed degradation algorithm used when the target BS does not have enough resources. Finally a summary is given.

3.2.1 Bandwidth Allocation Algorithm

Allocating bandwidth to new users is a relatively easy mission. The general rule the BS should follow is whether or not there are enough resources. Allocating bandwidth to handoff users is more complicated because the BS wants to maintain the user experience enjoyed in the source cell. The general idea is that the BS checks the information stored in the *User Profile* before making decisions.

Once the BS receives the connection request from an unknown user regardless of wether or not it is a new user or a handoff user, the BS first checks the validation of this user (this is discussed in *User ID* in Section 3.1.1 *The Structure of The User Profile*). Then the BS has to decide if this user is a new user or a handoff user. This requires the BS to send the inquiry to the MSC, which will check all *User ID*s within its controlling coverage area. If the MSC finds this *User ID* is currently managed in a cell, apparently this user is a handoff user. Otherwise, it might be a new user or a handoff user roaming from a cell controlled by another MSC.

Here we should introduce the difference between the Inter-MSC handoffs and the Intra-MSC handoffs shown in Figure 1.1. An MSC manages a number of cells; and a number of MSCs have a upper level management equipment. An Inter-MSC handoff means that the source cell and the target cell are controlled by different MSCs; whereas an Intra-MSC handoff denotes that the source cell and the target cell are managed by the same MSC. In this thesis and its simulation systems, for simplicity, we only study the Intra-MSC handoffs

though our approach can be extended to the Inter-MSC handoffs.

If the MSC cannot find the *User ID* within its controlling coverage area, the MSC has to check other MSCs to decide because this use might come from a cell managed by another cell. Therefore, the MSC checks other MSCs via its upper level management equipment. If the upper level management equipment finds this user is currently managed by another MSC, obviously this user is also a handoff user. Otherwise, this user can be identified to be a new user. The following pseudo-code illustrate this process.

```
01  //HOWTO decide a user is a new user or a handoff user;
02  BS gets a connection request from a user;
03  BS checks the validation of this User ID;
04  //detailed process is discussed in ''User ID'' in Section 3.1.1
05
06  BS sends inquiry to MSC;
07  MSC checks all users within its coverage area;
08  if (MSC finds this user is currently managed by a cell)
09      this user is a handoff user;
10  else{
11      MSC sends inquiry to its upper level management equipment;
12      the upper level checks all users within its coverage area;
13      if (this user is found within the coverage area of an MSC)
14          this user is a handoff user;
15      else
16          this user is a new user;
17  }
```

After deciding the user is a new user or a handoff user, the BS starts the corresponding process to handle the new user or the handoff user.

Allocating Bandwidth to New Users

Before admitting a new user, the BS has to check whether or not it has enough bandwidth to meet the user's initial bandwidth. As we discussed before, different classes of service correspond to different initial bandwidth. When a *User ID* is identified to be valid, the information of its class of service is acquired.

Since the carrier promises to guarantee the initial bandwidth the subscriber pays for monthly, the BS has to monitor whether the initial bandwidth can be ensured even if the resources currently consumed by a user is less than the initial bandwidth. For example, the initial bandwidth for a class of users is 150 Kbps. If this user is now consuming 100 Kbps, the BS still has to reserve another 50 Kbps (i.e. 150 Kbps in total) for this user to ensure the carrier's promise, because this user might suddenly ask for 150 Kbps to use in next time slot. To find the actual available bandwidth that the BS can allocate, the BS finds the larger of the reserved bandwidth and the summation of currently consumed bandwidth of all admitted users. The following definition indicates the *actual* available bandwidth that the BS can deploy.

Definition 1 *Let B_c be the capacity of the cell. Let B_{iu} denote the initial bandwidth required by a user. Let B_r be the total reserved bandwidth to ensure the initial bandwidth for all admitted users. Let B_{ua} be the summation of bandwidth consumed by all admitted users. Let B_{aa} denote the actual available bandwidth in the BS. Thus, there exists the following relation:*

$$B_r = \sum B_{iu} \tag{3.1}$$

$$B_{aa} = B_c - max(B_r, B_{ua}) \tag{3.2}$$

As Equation 3.2 shows, the BS compares the summation of initial bandwidth of all admitted users (i.e. reserved bandwidth for all admitted users) with the summation of actually consumed bandwidth of all admitted users. Then, the bigger one is used to compute the actually available bandwidth.

Once receiving the request from a new user, the BS will check where B_{aa} is enough to meet the initial bandwidth of this user. If true, the BS admits the user and builds the tuple of the *User Profile* for this user. At this moment, the BS assign *zero* bandwidth to the user because the user has not yet launched any application to access the Internet. However, the BS still has to reserve the initial bandwidth for this user lest this user requests the initial bandwidth in the next time slot. The *User Profile* in this case is illustrated in Table 3.2.

We have to note that even if the assigned bandwidth at this moment is *zero*, we still have to check whether or not the available bandwidth is equal to or larger than the initial bandwidth before admitting the user. There are two reasons as follows:

1. The initial bandwidth is the lower bound of resources that the carrier promises to provide to the user's RT connections. It is the provision of the contract between the users and the carrier and the users pay the monthly fee based on this.

2. Without checking this condition, the BS might not meet the first request from an admitted user. Suppose a new user is admitted before the BS ensures there is the initial bandwidth. After admission, the first request of this user is to launch a real time connection and this connection needs the bandwidth that happens to be equal to the initial bandwidth. In this case, an admitted user cannot be guaranteed to have the least promised bandwidth, which breaks the contract between the user and the carrier.

If there is not enough bandwidth to meet the initial bandwidth requirement, or the resources are exhausted, it is time for the BS to start the degradation process to handle the user (this is discuss in Section 3.2.2). The following pseudo-code illustrates the process of how to allocate bandwidth to a new user.

```
01  //HOWTO allocate bandwidth to a new user
02  //a user is identified to be a new user already ...
03  if (B_aa ≥ the initial bandwidth) {
04      BS admits the user;
05      BS builds a tuple of the User Profile for this user;
06      //the user actually consumes zero bandwidth now
07      //but BS still has to reserve the initial bandwidth for this user
08  }
09  else
10      BS starts degradation algorithm to try to admit this user;
```

Allocating Bandwidth to Handoff Users

Allocating bandwidth to the handoff user is different from allocating bandwidth to the new user, and it needs help from the *User Profile*. The purpose of the *User Profile* is to accurately describe the QoS requirements so that the target cell can use it to allocate the bandwidth for the handoff users. As a result, the target cell knows the QoS requirements in advance and can tailor its resource allocation to the level of applications. Thus, the handoff user might enjoy the same experience as it did in the source cell.

Consider the example in Figure 3.1 and suppose the user is *User 1* in Table 3.1. Hence, in the source cell, the user with class B is currently running two applications to access the Internet: one is a real time application currently consuming 0.3 Mbps, and the other is a non-real time application currently using 0.1 Mbps.

During the process when the user arrives at *P2* from *P1*, the corresponding tuple of the *User Profile* for *User 1* has already been forwarded from *BS1* to *BS4* via the *MSC* (how to forward the tuple of the *User Profile* is discussed in details in Section 3.1.3 *How to Apply the User Profile to The Handoff User*). Thus, *BS4* knows the requirements of QoS before it starts to allocate resources for the user. This makes it possible to maintain the same QoS experience for the user.

When considering the admission request from *User 1*, *BS4* first considers whether or not the BS can maintain the same user experience for the handoff user, i.e. whether or not the B_{aa} is equal to or more than the bandwidth currently consumed by the user. If true, the BS simply admits the user by allocating the corresponding bandwidth to this user. In the example in Table 3.1, if the B_{aa} is at least 0.4 Mbps, the BS directly admits this user and assigns 0.4 Mbps to this user. Due to *Rule 3.2*, *BS4* dedicates 0.3 Mbps for the real time application and allocates 0.1 Mbps tentatively for a non-real time application. We should note that the bandwidth assigned to the non-real time connection can be taken back by the BS during the process of degradation (detailed information can be found in Section 3.2.2). This case is the best case, where the user experience is perfectly maintained and the QoS

requirements are entirely met.

The second case is when the B_{aa} is less than the bandwidth currently consumed by the user but more than the initial bandwidth. For example, the user currently consumes 1.5 Mbps and the initial bandwidth of this user's class is 1 Mbps. The BS has to analyze the composition of the currently consumed bandwidth which has the following two subcases:

1. The bandwidth used for the RT connections is more than the initial bandwidth. In the above example, suppose the bandwidth consumed for RT connections and NRT connections are 1.1 Mbps and 0.4 Mbps respectively. In this case, the BS has no ability to admit the user based on its QoS requirements, i.e. to supply 1.1 Mbps to allow the running RT connections. Instead, the BS only can admit the user according to the initial bandwidth. Thus, the BS admits the user and dedicates the initial bandwidth (which is 1.0 Mbps in this example) to the user's RT connections and *zero* bandwidth to its NRT connections. This means the BS only can follow the signed contract between the carrier and the user to guarantee the initial bandwidth. Obviously, the user's RT connection degrades its quality but the user is successfully admitted and the BS has met the guaranteed requirement of the RT connections.

2. The bandwidth used for RT connections is less than the initial bandwidth. For example, the user is consuming 0.9 Mbps for RT connections and 0.6 Mbps for NRT connections. In this case, the BS admits the user and then dedicates the corresponding bandwidth to RT connections, and allocates resources to NRT connections so that the total bandwidth assigned to the user is up to the initial bandwidth. Thus, in this example, the BS allocates 0.9 Mbps to RT connections and 0.1 Mbps to NRT connection. Like the above case, the BS first admits the user and then does it best to minimize the future impact of RT connections.

The final case occurs when the B_{aa} is less than the initial bandwidth of the RT connections, the BS cannot guarantee the lower bound of the bandwidth the carrier promises

in the contract. The BS now triggers the degradation process to *try* to admit the user, instead of directly rejecting the user at this moment. If there is enough bandwidth after the degradation, the user is still admitted. Otherwise, the user has to be rejected.

The rule we should use in this case is generated as follows.

Rule 3.5 *Both handoff users and admitted users can be assigned more bandwidth than their initial bandwidth if and only if there exists large enough B_{aa} and the request is not larger than an upper bound, which is subject to different classes.*

A tricky case we should consider is that if multiple users, including the handoff users and new users, simultaneously asks for the limited resources and only some of them could be admitted based on the current value of B_{aa}. *Rule 3.1* again should be obeyed in this case. At first, the BS sorts the users based on their classes. Then the BS admits them from the highest class until the bandwidth is exhausted. At this point the degradation process (discussed in detail in next section), which is also class-based, frees up more resources. The following essential rule is that fairness must be kept, in which case the admission process is class-based and different classes result in different treatment. It is fair for users who pay a higher monthly fee to be treated with higher priority.

Rule 3.6 *Following the Rule 3.1, the BS admits multiple users in a class-based manner to allocate bandwidth to users with the highest class. If the bandwidth is exhausted, the BS starts the degradation process to use the same way to admit users.*

The following pseudo-code describes the entire bandwidth allocation algorithm for the single user in the ideal condition and for the multiple users with various classes in the practical condition.

```
01  //the bandwidth allocation algorithm for a single user
02  //after the target BS checks the forwarded User Profile
03  //       and the B_aa ...
04  if (B_aa ≥ currently used bandwidth of this user) {
05      //do not care if the request is beyond the initial bandwidth
06      BS admits this user;
07      BS allocates bandwidth with the same composition as the user;
08      //so, user keeps the same user experience after handoff
09  }else {
10      if (B_aa ≥ initial bandwidth) {
11          if(used bandwidth for RT connections
12                      ≥ initial bandwidth){
13             BS admits the user;
14             BS dedicates initial bandwidth only to RT connection
15                   and allocate zero to NRT connections;
16          } else {
17              BS admits the user;
18              BS dedicates the same bandwidth as the user has to
19                     RT connection, and allocate some resources
20                     to let the user enjoy the initial bandwidth;
21          }
22      } else {
23          //BS cannot guarantee the initial bandwidth now
24          //BS starts degradation process to try to admit the user;
25          BS starts degradation process to take back some bandwidth;
26          if (chopped resources meet the user's request) {
27               BS admits the user;
28               BS allocates initial bandwidth to the user;
29          } else
30                //BS has no choice now
31                BS rejects the user;
32      }
33  }
```

```
01  //the bandwidth allocation algorithm for the multiple users
02  //after the target BS checks the forwarded User Profile
03  //       and the B_aa ...
04  if(there is enough bandwidth)
05      sort all requests based on their classes;
06      while(there is still B_aa)
07          admit the user with the highest class;
08      if(there still exist users) {
09          //now there is no bandwidth left
10          start the degradation process to chop some resources;
11          while(there is still enough chopped resources left)
12              BS admits the user with the highest class;
13      }
14  }
15  BS rejects all requested users;
```

Allocating Bandwidth to Admitted Users Who Ask for More Bandwidth than Their Initial Bandwidth

How to handle an admitted user is different from admitting a user. Once the user establishes a connection, the BS assigns the bandwidth to the user and accordingly updates the *User Profile*. However, more details should be discussed. In our mechanism, we allow the user to use more bandwidth than the initial bandwidth. We think that the resources are wasted if the request of this kind is not allowed when there is still B_{aa}. However, we also cannot allow the user to freely and greedily request the resources without any constraints because this obviously will make the system resources be exhausted easily. In cases when there are a number of malicious resource-hog requests, the system is fairly vulnerable.

We define the bandwidth upper bound (BUB) to constrain the degree of freedom of asking for more resources. The BUB is a multiplication coefficient which is subject to the different classes. It is designed to indicate how much bandwidth a user can consume above its initial bandwidth. Based on *Rule 3.1*, we believe the higher class should have higher priority and higher flexibility, i.e. the higher class users are allowed to use relatively larger BUB. For example, BUB for the lowest class users is 1.0; it is 1.5 for the intermediate

class and 2.0 for the highest class. Thus, the actual upper bound of the bandwidth every class users could use in practice is $1.0 \times (initial\ bandwidth)$, $1.5 \times (initial\ bandwidth)$ and $2.0 \times (initial\ bandwidth)$, respectively, as class priority increases. The rule we must follow is that an admitted user is allowed to ask for more bandwidth if and only if (a) there is B_{aa} and (b) its request is not beyond the predefined upper bound B_u, which is given in the following definition.

Definition 2 *Let B_i be the initial bandwidth subject to a class. Let B_b be the bandwidth upper bound subject to a class. Then B_u is the largest actually used bandwidth for a class.*

$$B_u = B_i \times B_b \tag{3.3}$$

Therefore, the BS allows the consumed bandwidth to be greater than the initial bandwidth in this limits by checking whether or not the total consumed bandwidth, after the assignment, is more than B_u. If the summation of all connections for a user is lower than the above constraint, the request is admitted as long as there is B_{aa}. The following pseudo-code illustrates how the BS assigns the bandwidth for the admitted user.

```
01    //HOWTO allocate allocate bandwidth to the admitted user
02    //an admitted user requests more bandwidth
03    if (there is enough bandwidth ) {
04        if (after assignment, the totally consumed bandwidth > Bu)
05            BS only assigns Bu to the user;
06        else
07            BS assigns the requested bandwidth to the user;
08    }
```

3.2.2 Bandwidth Degradation Algorithm

The critical purpose of the degradation algorithm is that the BS must do something instead of directly rejecting the user when a BS does not have enough available resources to admit users. Asking the system with exhausted bandwidth to admit the users is seemingly a paradox but it is actually possible when recalling the nature of NRT traffic.

The metric of NRT records the bandwidth consumed for non-real time applications, whose characteristic is that the allocated bandwidth is not required to be maintained continuously because the data delivery in the connection is not continuous. Furthermore, it has high delay tolerance. Therefore, the allocated resources for the real time application must be held during the entire transmission time.

The degradation algorithm is based on this thinking: now that the NRT application is able to tolerate time delay, we might *temporarily borrow* the bandwidth assigned to the NRT connection to admit some other users and then *return* the bandwidth back to the user. In the above example, if the BS takes back the previously assigned bandwidth 100 Kbps when the user is sending the email, the BS can admit more users.

One thing pertaining to RT connections should be clarified here. Based on *Rule 3.5*, a mobile user is allowed to consume more bandwidth than its corresponding initial bandwidth if there are available resources and its bandwidth request does not break the constraint of BUB. Thus, the bandwidth consumed for RT connections of a mobile user might also be changeable: it might be more than initial bandwidth and therefore might be reduced to initial bandwidth in the degradation process (this will be discussed in detail later in this section). In general, the bandwidth consumed by RT connections is more stable than that consumed by NRT connections.

In the process of the degradation, we still should follow the principle of *Rule 3.1*, which indicates that the degradation should start from the lowest class to the highest class due to the different priority. In practice, we at first sort all currently admitted users on the basis of classes and the assigned bandwidth for the non-real time applications. That means we sort all users based on the currently consumed bandwidth for all NRT applications running in one user of the lowest class, the intermediate class and the highest class, respectively. If we take Table 3.1 as the example, the sorted result should be like Table 3.6. As we can see, the sorted list begins from the lowest class C and in every class, the users are sorted based on the bandwidth for all non-real time applications of one user.

Table 3.6. A SORTED DEGRADATION LIST BASED ON BANDWIDTH CONSUMED BY NRT CONNECTIONS

ID	Class	Σ
User 3	C	0.5
User 1	B	0.1
User 4	A	0.8
User 2	A	0.0

Then, we begin with the users in the lowest class to take back the assigned bandwidth for NRT applications. After degrading a user, the BS checks whether or not the summation of the originally left bandwidth and the currently chopped bandwidth is enough to admit this user. As we explained before, "enough bandwidth" in this context means at least initial bandwidth to new users or enough B_{aa} to handoff users (more detail explanation can be found in previous *Allocating Bandwidth to New Users* and *Allocating Bandwidth to Handoff Users*).

If the BS does not have enough bandwidth to admit the request, the degradation will continue until the user is admitted or all admitted users are degraded. For example, we first degrade 0.5 Mbps from *User 3*. Then, we check whether or not the summation of left bandwidth and this 0.5 Mbps can admit the user. If true, the BS admits the user; otherwise, the BS takes away the bandwidth of the next user in the table, i.e. 0.1 Mbps is taken away by the BS from *User 1*. This process will continue until the requesting user is admitted or all users in the above table have had their corresponding resources for the NRT connection taken away. If all users in the above sorted list are degraded and the summation of bandwidth is still not enough to admit the user, the BS starts to consider the bandwidth consumed by RT connections.

Reducing bandwidth consumed by RT connections to the initial bandwidth is based on the rule of fairness. Based on *Rule 3.5*, a mobile user is allowed to consume more bandwidth for RT connections than the initial bandwidth if the bandwidth is available. For instance, a user can consume 0.5 Mbps for its RT connections whereas the initial bandwidth

of this user is 0.384 Mbps. This approach enables the BS not to waste the bandwidth. However, it is not fair to handoff users in the case, where the resources are exhausted and a handoff user asks for the admission. Thus, when the BS still does not have enough bandwidth to admit a handoff user after degrading all NRT connections, the BS cannot allow the users to consume more bandwidth than the initial bandwidth anymore. In practice, the BS first sorts *User Profile* based on the bandwidth consumed by RT connections as Table 3.7. As we can see, this table begins from the lowest class C and in every class, users are sorted according to the bandwidth for all RT connections of one user.

Then the BS, starting at the lowest class, checks the consumed bandwidth for RT connection of every admitted user, and degrades the consumed bandwidth to the initial bandwidth if the consumed bandwidth is beyond the initial bandwidth. To the Table 3.7, the BS will only allow *User 3* to consume 0.144 Mbps for RT connections and 0.056 Mbps is saved. Then the BS checks whether the updated available bandwidth is enough to admit the handoff user. If true, the BS admits this handoff user; otherwise, the BS checks the next admitted user *User 1*. This process will continue until the requesting user is admitted or all admitted users have had their *extra* resources for the RT connections taken away. If all admitted users are degraded and the summation of bandwidth is still not enough to admit the request, it is rejected.

Table 3.7. A SORTED DEGRADATION LIST BASED ON BANDWIDTH CONSUMED BY RT CONNECTIONS

ID	Class	Σ
User 3	C	0.2
User 1	B	0.4
User 2	A	1.5
User 4	A	0.7

The degradation process is triggered if and only if the bandwidth is exhausted. Since the worst time cost of the degradation process occurs when is all admitted users in the cell have to be degraded, we should be careful about starting the degradation process. The

values of *BUB* discussed in the previous section indirectly affects the degradation process. *BUB* denotes how greedily the user can consume the resources. Larger *BUB* allows the system to run out of resources faster; while smaller *BUB* plays the contrary role. We also discussed in the previous section that the value of the *BUB* could be parameterized. Thus, the frequency of the degradation can be adjusted by controlling the value of the *BUB*.

Rule 3.7 *The allocated bandwidth for a non-real time application and the extra bandwidth of RT connections could be taken back by the BS if the resources are exhausted.*

The following pseudo-code describes the entire degradation algorithm.

```
01  //the bandwidth degradation algorithm
02  boolean admission = false;
03  if(there is not enough bandwidth) { //OUT-IF
04      //start the degradation algorithm
05      sort all users based on the class and the sum of used
06                bandwidth for NRT connections;
07      float x = currently available bandwidth;
08      float y = 0.0f;
09      float sum = 0.0f;
10      for each user in the sorted list {
11          y = chopped the bandwidth of the first user;
12          sum += x + y;
13          if (sum is enough to admit the user) {
14               admit the user;
15               //stop the degradation
16               admission = true;
17               break;
18          }
19      }//end of for
20      if (admission == false) { //IN-IF
21          sort all users based on the class and the sum of used
22                    bandwidth for RT connections;
23          for each user in the sorted list {
24              if (the used bandwidth for RT connections
25                            > initial bandwidth) {
26                  y = currently used bandwidth for RT connections
27                            - initial bandwidth;
28                  set the used bandwidth for RT connections of this
29                            user to be initial bandwidth;
30              }
31              sum += x + y;
32              if (sum is enough to admit the user) {
33                   admit the user;
34                   //stop the degradation
35                   admission = true;
36                   break;
37              }
38          }
39      } //end of IN-IF
40  }//end of OUT-IF
41  if(admission == false) {
42      //still have no enough bandwidth after degrading all users
43      reject the user;
44  }
```

3.2.3 Summary and Discussion

In this section, we discuss our bandwidth allocation algorithm and the bandwidth degradation algorithm. Each of them has some attractive features.

The first advantage of our bandwidth allocation algorithm is that the *current level of QoS is ensured as much as possible after handoffs*, so that the user experience can be most likely kept after handoffs. For the first time the system has some way to store the original requirements of QoS in advance in the *User Profile* and then the target BS takes advantage of this information to allocate the bandwidth.

In the meantime, the notion of *doing the best* is demonstrated in the algorithm. Due to the limited resources, it may be impossible to admit every requested user, including the handoff users and the new users. Based on some rules introduced in this section, the target BS does its best to admit the users. When allocating the bandwidth, the BS should provide for the users if there are available resources. However, to avoid a user requesting too much bandwidth, we define a coefficient BUB to limit the upper bound of resources the user could consume. For a handoff user, the BS first compares the user's currently used bandwidth with the B_{aa}, then compares the bandwidth consumed for RT connections and the initial bandwidth. Obviously, the BS retreats to the lower level once the current aim cannot be achieved. Finally, even if the B_{aa} is less than the initial bandwidth, the BS launches the degradation process to *try* to admit the user by reducing usage of some bandwidth. If and only if the recovered bandwidth still cannot admit the user's requirements, this user is rejected. Thus, the best effort is shown in every step during the entire process of the admission and the bandwidth allocation.

This approach also follows *Rule 3.1*, the expression of fairness. In the bandwidth allocation algorithm, the notion of *fairness* is expressed in three places. One place is that different classes mean different priorities of admission, which lets a BS differently treat users when the limited resources only allow some of the users to be admitted. In this case, the best effort is done by the BS to admit users on the basis of classes. Even if during the

process of the degradation, the BS still uses the chopped bandwidth to permit admission based on classes. The second place where we can realize the *fairness* is that the BUB is defined for each class. A higher class service allows more flexibility and accordingly higher BUB so that users of this kind can enjoy more bandwidth than the initial bandwidth. In contrast, a lower class acquires the lower BUB so that they are only allowed to ask for a bit more than the initial bandwidth. The final place we encounter the *fairness* is the first part of our *Rule 3.5*. The monthly fee defines the initial bandwidth a user is allowed to enjoy, but should it ask for more resources while there exists available bandwidth? If not, the available resource is wasted and the user experience decreases greatly, in which case the *unfairness* is significant. If yes, users might be concerned that other users are *resource hog*, who greedily asks for as much extra resources as possible. Then, the system is inevitably busy in running the degradation algorithm to chop resources continuously. The worse case is that some malicious users might intentionally use this feature to let the system be in a constant busy state. But by using our created notion of BUB, the problem of a resource hog could be easily avoided. The level of greed is limited within a certain controllable degree. Thus, the *fairness* of this kind can easily be kept without negative influence.

Therefore, our bandwidth allocation algorithm actually has another feature *hog-avoidance* benefited from the notion of ***BUB***. The resources hog is avoided by the predefined upper bound of requested resources. In addition, this *hog-avoidance* is *parameterized*. By modifying the value of ***BUB***, people can pre-control the level of greed of the use of extra bandwidth. If a cell in a suburban area is not busy in the year of the installation of equipment, people could allow relatively large value of BUB so that the users, whose number are small at this time, can ask for more extra bandwidth. However, if this cell became more and more prosperous after a period of time, people can modify the value of BUB to be smaller so that the users, whose number is much larger than before due to the popularity at this time, are allowed to ask for only a bit more than the initial bandwidth, or even the same as the initial bandwidth.

As to our degradation algorithm, one most important feature is to *efficiently use the resources*. By analyzing the characteristic of the assigned bandwidth for the NRT application, we realize that the NRT connections have high delay tolerance and the required bandwidth is actually *zero* in many cases in the entire transmission duration. Thus, we believe that the BS should efficiently assign resources, i.e. BS can temporarily take back the bandwidth consumed by an NRT connection, and assign this chopped resource to admit other users.

The above analysis also indicates another feature that is the *best effort admission*. The sufficient condition of our degradation algorithm is that there is not enough bandwidth. In this case, people have two choices: one is to passively keep the current situation to do nothing and directly reject the user; the other is to actively reserve some potential resources to admit this user. Every approach has advantages and disadvantages. The first choice of course lets the system not do any extra work. Thus, the extra system overhead on the BS side is *zero*. But its drawback is similarly significant: it actually does not admit this user. Hence, the admission ratio in this mechanism is very low. In contrast, the second approach does its best to admit the user so that it has a much higher admission ratio, which is shown by the simulation discussed in the next chapter. But there is extra system work from time to time, i.e. the BS is much busier than the BS in the first case.

The question of "*which approach is better*" is actually the question: which one, between the busier BS and higher admission ratio, is more important? We believe that the higher admission ratio might be more important than the former one though the busier system might affect the user experience, because the user experience is the most important thing for both the carrier and the clients. Lower admission ratio means a high possibility for the handoff users to be rejected, which means surfing the Internet or the checking of email behavior is easily interrupted when the user is roaming in the street. This might be very annoying for both users and the service providers. Thus, in consideration of the user experience, we prefer the degradation algorithm instead of directly rejecting users to improve the admission ratio.

Another feature of our degradation algorithm is *trying to keep the same user experience* to increase the admission ratio. The degradation process degrades the bandwidth from users, which seems to degrade the user experience because the stereotype idea is that the user experience is subject to the bandwidth – the higher bandwidth means the better experience. Because we accurately divide the connections into RT and NRT connections, the stereotype idea is always true for the RT connections but might not be true for the NRT connections, since the NRT connections can tolerate the high time delay. On the basis of this understanding, degrading bandwidth from NRT connections has the least impact on the user experience.

The degree of the use of the degradation process is *controllable and parameterized*. By adjusting the value of the BUB, we control the degree of the exhausted system resources, which further decides how easily the degradation process is triggered. The larger BUB is, the more greedy the users could be, the more easily the system resources are exhausted, and finally the more easily the degradation is triggered. Indirectly, the easiness of the degradation is controlled and parameterized by this.

3.3 The Entire Working Process of the Source BS

In this section, we combine all concepts so far together to give an example to show the entire cycle of the working process of the source BS. The state machine in Figure 3.2 illustrates the change of working states, with respect to a particular MS, of the source BS. An arc in Figure 3.2 refers to an event transitions and a circle is the state of a BS.

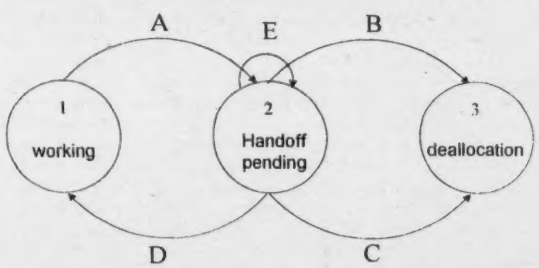

Figure 3.2. THE FINITE STATE MACHINE FOR A WORKING SOURCE BS

The detailed discussion is as follows:

1. *State 1* is the ordinary working state, where the MS is closely moving around the source BS. At this moment, the source BS manages the "last mile" of the connection of the MS: the source BS receives the data from the MS and forwards them to MSC, and this BS also delivers data from the MSC to the MS. The source BS keeps the *User Profile* of this MS in its memory and the MSC at this time does not know this *User Profile* because the source BS does not yet forward the *User Profile* to the MSC. The target cell are not yet involved in this state.

   (a) *Event A* is the case where the system notifies the source BS that the MS enters to the handoff state. As a result, the source BS switches into *State 2*.

2. *State 2* is actually a ready state and an intermediate state, in which case the source BS and the MSC start to do some preparation for the handoff. This involves that the source BS forwarding the *User Profile* to the MSC, and subsequently the system starting its selection process to select a target BS. In this state, although the system selects the target BS, it still does not yet send the instruction of naming the target BS to the source and target cell. Thus, the source BS still holds the connection with the user.

   (a) *Event D* means that the MS in the boundary of cells moves back towards the source BS. Since the MS receives stronger and stronger signal from the source BS, the source BS is selected again to continuously manage the connection of the MS. Precisely, there are two situations in this case as follows:

      i. If the MSC already determined the target BS in *State 2*, the system will again realize the original source BS receives stronger signal because the user is currently moving towards the source BS. Therefore, the original source BS is again chosen to be the controlling BS.

ii. If the MSC has not yet determined the target BS in *State 2*, the system will notify the source BS to be still responsible for controlling the MS because of the apparent trend of strength of signal.

In any case, the source BS will *still* manage the MS. Hence, there is no need anymore for the MSC to maintain the *User Profile* forwarded from the source BS in *Event A*. Thus, the MSC removes the *User Profile* from its memory.

(b) *Event B* means that the MSC finally sends the instruction about the selection of the target BS to both the source BS and the target BS. This event lets the target BS start the associated bandwidth allocation process to admit or reject the MS. After finishing this work, the target BS informs the source BS via the MSC the completeness of the admission process. Once getting this information, the source BS switches to the *State 3*.

(c) *Event C* refers to the case when a user at the boundary of cells stops the conversation by itself due to either the user's operation or low battery. As a result, the source BS enters the *State 3*.

(d) *Event E* refers to the case when a user at the edge of a cell roams around the boundary of cells. The user might enter another cell, but not go much further before it comes back to the original cell. In *Event E*, the MS can hear signals from multiple BSs and the system is forced to continuously select the target BS. But the source BS still holds the connection with the MS and the system does not *yet* explicitly assign the target BS.

3. *State 3* is the termination state for the source BS with respect to the MS, in which case the source BS closes the connection with the user and releases the resources used by this disconnecting user so that other users can re-use such resources.

3.4 Summary

This chapter explains in detail our approach to effectively and efficiently allocate bandwidth to handoff and new users. Our solution consists of two parts: the bandwidth allocation algorithms and the bandwidth degradation algorithm. The former is to allocate bandwidth to handoff and new users in consideration of QoS issues. The latter is to admit handoff and new users if the resources are exhausted.

In terms of data services, keeping connection with a roaming MS is not enough. Users normally still have requirements for QoS, i.e. how to make the target BS timely treat the handoff user in the same way as what the user experiences in the source cell. To let the BS understand the QoS requirements of every connection of every user, we defines the *User Profile* to store such information.

The *User Profile* enables the bandwidth allocation to be *class-based* and *application-aware*. It uses the least data to clearly record the important metric requirements for QoS for every connection. While the *User Profile* follows the widely accepted rule (*Rule 3.1: The class of service indicates the priority of the service, which is used for class-based treatment.*), which gives higher priority to higher class, it further studies QoS up to the level of the application, which distinguishes RT and NRT applications. This improvement of granularity describes the QoS requirements more precisely. In addition, the differentiation between RT and NRT applications makes it possible for our degradation process later.

The *User Profile* enables the entire system to have a reliable way to record and store QoS status and requirements, and also makes it possible for the handoff user to keep the same experience after handoff. Thus, for the first time, the system has some way to know the applications running on the client side and the information of bandwidth consumed by the user at the level of applications. This makes it possible for the target BS to allocate resources to handoff users at the level of the *application* instead of the fuzzier notion of a *user*. The granularity is improved greatly in this area. We think that knowing the precise information of QoS requirements is the premise of keeping the same QoS after handoffs.

Finally, the notion of *class-based* is still kept in our approach. The class of services not only denotes the different service and different initial bandwidth, but also is used by the BS to handle handoff users according to class priorities.

Chapter 4

Simulation Systems

In our thesis project, we develop two simulation systems to compare the performance difference between our bandwidth allocation mechanism and the conventional mechanism. One simulation system emulates the ordinary system, whose BS has no notion of RT/NRT connections and directly rejects users when the resources are exhausted. The other simulation system runs our new mechanism of bandwidth allocation and actively deals with users by the degradation process in case the bandwidth is exhausted.

The simulation is an event-driven system. The simulation model is a single target cell. The incoming handoff users and new users belong to the arrival event. The bandwidth allocation measurement is applied for each arrival. After a certain period of time, called *Duration Time*, the user will leave the cell, which accordingly triggers the leave event. To mimic the real world as much as possible, the simulator has the dynamic consumed bandwidth, i.e. the simulator periodically changes the bandwidth consumed by each admitted user. In particular, we are interested in average admission performance and bandwidth utilization.

The two simulation systems are Windows-based applications and they have user-friendly interfaces. They are designed to give users some flexibility, including the way to setup parameters and the way to store the generated data.

This chapter consists of four parts. We first describe our simulation model in detail in Section 4.1.1, including the model itself, an example showing how the simulator works and

the feature of changing consumed bandwidth for every admitted user. Then we move to the topic of the development environment in Section 4.2. In Section 4.3, we discuss the verification of implementation and validation of the model. Finally, there is a summary. The experimental results and analysis are in Chapter 5.

4.1 Simulation

Two simulation systems are developed and each simulator has the same model but different bandwidth allocation mechanisms. The simulator running the traditional bandwidth allocation algorithm is named *ThesisBaseCase*; the simulator running the novel mechanism is called *ThesisMyCase*. Since our simulators are parameterized, a set of parameters can be input to the systems; and then by comparing the results of different systems, we are able to evaluate the performance of different mechanisms used in the different systems.

This section focuses on the simulation modelling and its related design. We first explain the modelling we use in the simulation, then followed by what parameters the simulators required and what metrics we use to evaluate the performance. Finally, we will briefly introduce our software development and the verification and validation.

4.1.1 Simulation Modelling

The Model

A single target cell is modelled. After carefully analyzing the various relations among the handoff users, new users and the BS, we decide to study the generic scenario consisting of a target cell for all admission and management work.

Based on [30, Chapter 30], the arrival rate λ (including incoming handoff users and new users) follows a Poisson distribution, and the interarrivals follow an exponential distribution. The duration time μ (regardless of class) follows an exponential distribution as well and represents the length of time the users will remain in this cell. Both the arrival rate λ and the average duration time μ are user-defined parameters.

This event-driven simulation consists of the following events including *HandoffEvent*, *NewUserEvent*, *LeaveEvent*, *LogEvent* and *PeriodicBWChangeEvent*. Both *HandoffEvent* and *NewUserEvent* are actually arrival events. Straightforwardly, upon a handoff, a user comes into the target cell, the *HandoffEvent* is fired; once a new user sends an admission request, the *NewUserEvent* is triggered; and when any user leaves the target cell or stops its communication, the *LeaveEvent* is generated. The *LogEvent* is the event triggering the simulator to log the current system metrics into the disk. Once some event is fired, the simulator will be triggered to do something as discussed below:

1. For the *HandoffEvent*, the simulator does the following:

   (a) The *SystemTimer* is set to the time of this arrival event.

   (b) An instance of class `User`[1] is created and its data member `Type` is set to be `Handoff`. In addition, the simulator generates the characteristics of this user, including the bandwidth currently consumed for RT connections and NRT connections, respectively.

   (c) The BS tries to admit this user based on the traditional or our novel admission mechanism; if the BS cannot admit this user, the BS rejects this user and the simulator updates the metric *RejectHandoffs* which is the number of rejected handoff users.

   (d) If this user is admitted, the metric *AllowHandoffs* is updated. This metric records the number of admitted handoff users.

   (e) If this user is admitted, the function `DurationExp` is invoked to know the duration time of this handoff user in the target cell and accordingly calculate the leave time of the *LeaveEvent* for this user. Then the `TimeStamp`[2] of this user is set to (*SystemTimer* + `DurationExp()`).

[1] We have a class `User` to describe both handoff users and new users. Its data member `Type` indicates the instance is either a handoff user or a new user.

[2] The class `User` has a data member `TimeStamp` which denotes the leave time of the user.

(f) The function `NextHandoffExp` is invoked to know the interval time between the current arrival handoff user and the next coming handoff user, which is the time of next *HandoffEvent*. The *HandoffTimer* is set to (*SystemTimer* + `NextHandoffExp()`).

2. For the *NewUserEvent*, the simulator does the following:

   (a) The *SystemTimer* is set to the time of this arrival event.

   (b) An instance of class `User` is created and its data member `Type` is set to be `NewUser`. In addition, the simulator generates the characteristics of this user, including the bandwidth currently consumed for RT connections and NRT connections, respectively.

   (c) The BS tries to admit this user based on the traditional or our novel admission mechanism; if the BS cannot admit this user, the BS rejects this user and the simulator updates the metric *RejectNewUsers* which is the number of rejected new users.

   (d) If this user is admitted, the metric *AllowNewUsers* is updated. This metric records the number of admitted new users.

   (e) If this user is admitted, the function `DurationExp` is invoked to know the duration time of this new user in the target cell and accordingly calculate the leave time of this user. Then the `TimeStamp` of the *LeaveEvent* for this user is set to (*SystemTimer* + `DurationExp()`).

   (f) The function `NextNewUserExp` is invoked to know the interval time between the current new user and the next coming new user, which is the time of next *NewUserEvent*. The *NewUserTimer* is set to (*SystemTimer* + `NextNewUserExp()`).

3. For the *LeaveEvent*, the simulator does the following things:

   (a) The *SystemTimer* is set to the time of this leave event.

(b) The resources consumed by this user are released. Accordingly, two metrics *CurrentlyUsedBandwidth* and *CurrentUsersNumber* should be updated. The former one records all bandwidth currently used by all users. The latter records the current number of all admitted users in the range of the target cell.

4. For the *LogEvent*, the simulator logs a number of current system metrics. The metrics will be discussed in detail in Section 5.1 *Performance Metrics*.

5. For the *PeriodicBWChangeEvent*, the simulator periodically readjusts the bandwidth consumed by every user and we set this period to be 1 minute. Since the consumed bandwidth of each user is dynamically changed, i.e. the bandwidth might increase or decrease, the simulator, at each 1 minute interval, changes bandwidth by increasing or decreasing the consumed bandwidth randomly. How much the bandwidth can be increased will be limited by the value of *BUB* and the available bandwidth based on the algorithm in Definition 1 in Section 3.2.1.

6. Reset the time of next *PeriodicBWChangeEvent*.

In our simulation, we implement the following family of functions. Each of them takes the system time as the seed and takes the user-defined parameter as the argument to generate the random number.

1. There are two types of arrival rates: the arrival rate of the handoff user and the arrival rate of the new users. We let the simulator take a parameter *ArrivalRate* and a parameter pertaining to the ratio of the number of the handoff users to that of the new users, and then the simulator calculates the arrival rates for two types of users respectively (more details can be found in 5.2.2 *Parameters*). By using two arrival rates, the simulator invokes the functions as follows:

   (a) The function `double NextHandoffExp(double handoffInterArrival)`. It takes the interarrival rate of the handoff users as the argument and gives the ran-

dom number which is the interval time between the current arrival handoff user and the next handoff users and follows exponential distribution.

(b) The function `double NextNewUserExp(double newUserInterArrival)`. It needs the interarrival rate of the new users as the argument and returns the random number which is the interval time between the current new user and the next new user and follows exponential distribution.

2. The function `double DurationExp(double durationMu)`. This requires the duration time μ as the argument and gives the random number which is the duration time (i.e. the time a user stays in the cell) of a user and follows an exponential distribution.

The simulation has three different timers: the *SystemTimer*, the *HandoffTimer* and the *NewUserTimer*. The *SystemTimer* is the simulation timer and it is equivalent to the time tick of the target cell; the *HandoffTimer* is a timer specifically remembering the time of arrival and leave events of handoff users; and similarly, the *NewUserTimer* is the timer solely keeping track of the time of arrival and leave events of new users. The leave time are kept in a sorted queue.

An Example of The Working Process of the Simulation

Figure 4.1 illustrates an example showing the working process of this simulation from the very beginning. Although ordinarily there would be some *PeriodicBWChangeEvent* as well; we have ignored this for simplicity. For easier understanding, the handoff users and the new users are separately drawn: the handoff users are above the time axis and the new users are located below the time axis.

We can see from Figure 4.1(A) that the *SystemTimer* is initialized to 0 at the beginning. The function `NextHandoffExp` and `NextNewUserExp` are invoked and they return *H1* and *N1*. We suppose *N1* is less than *H1* in this example. This means the first arrival user to this

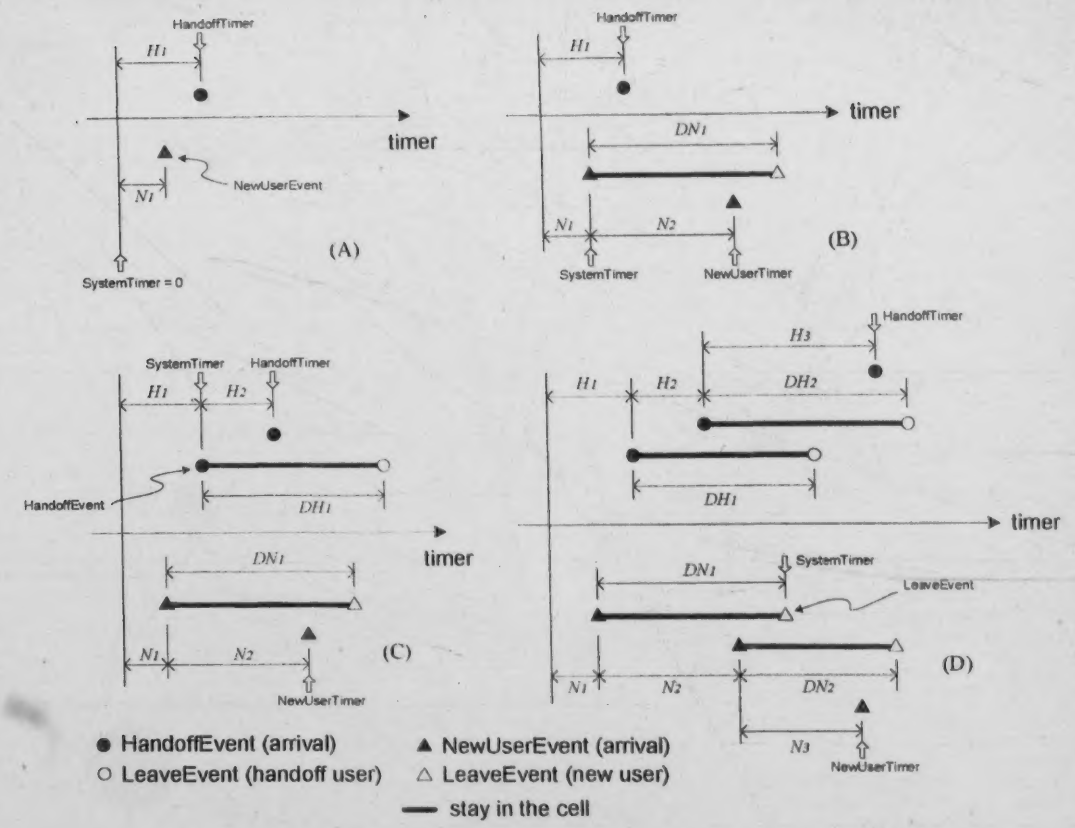

Figure 4.1. AN EXAMPLE OF THE WORKING PROCESS OF THE SIMULATION

target cell is a new user[3]. The *HandoffTimer* is set to *H1* and then a *NewUserEvent* is fired.

Figure 4.1(B) indicates that upon *NewUserEvent*, the *SystemTimer* is set to the time of this arrival event *N1*. The BS tries to admit this new user based on either the traditional or our novel admission mechanism. If the BS rejects this user, the *RejectEvent* is fired where the metric *RejectNewUser* is incremented by 1, i.e. one more new user is rejected. If this user is admitted, the metric *AllowNewUsers* is incremented by 1, which means one more new user joins the target cell. If accepted, the function `DurationExp` is invoked and returns *DN1* to get the duration time of this new user in the target cell and the leave time of this user is set to (*N1* + *DN1*). Thus the `TimeStamp` of this user is set to the leave time and

[3]Of course, *H1* could be less than *N1*, in which case the first arrival user to this target cell is a handoff user. How the simulator works the rest of time is no different from our example.

then this user is enqueued [4]. The function `NextNewUserExp` is invoked and returns *N2*, which is the interval time between the current new user and the next coming new user. The *NewUserTimer* is set to (*N1* + *N2*).

Now three timers have different values: *SystemTimer* is *N1*, *HandoffTimer* is *H1* and *NewUserTimer* is (*N1* + *N2*). There is one element in the queue whose *TimeStamp* is (*N1* + *DN1*).

At this moment, we should decide the next earliest event in the system. We compare the earliest leave time of the accepted users with *HandoffTimer* and *NewUserTimer*. In this case, the earliest time is *HandoffTimer* which is *H1*. Because *H1* is the arrival time of a handoff user, a *HandoffEvent* is fired.

Figure 4.1(C) denotes that upon *HandoffEvent*, the *SystemTimer* is set to *H1* and the BS tries to admit this handoff user. If the user cannot be admitted, the *RejectEvent* is fired where the metric *RejectHandoff* is incremented by 1, i.e. one more handoff user is rejected. If this handoff user is admitted, the metric *AllowHandoffs* is increased by 1, which means one more handoff user joins the target cell. The function `DurationExp` is invoked and returns *DH1* to get the duration time of this handoff user in the target cell and the leave time of this user is (*H1* + *DH1*). Thus the `TimeStamp` of this user is set to the leave time and then this user is enqueued. The function `NextHandoffExp` is invoked and returns *H2*, which is the interval time between the current handoff user and the next coming handoff user. The *HandoffTimer* is set to (*H1* + *H2*).

Now three timers have different values: *SystemTimer* is *H1*, *HandoffTimer* is (*H1* + *H2*) and *NewUserTimer* is (*N1* + *N2*). There are two elements in the queue: a new user whose *TimeStamp* is (*N1* + *DN1*) and a handoff user whose *TimeStamp* is (*H1* + *DH1*) respectively.

Again, we need to find the next earliest event based on the value of timers and the *TimeStamp* of elements in the queue. The earliest time at this moment is *HandoffTimer* (*H1* + *H2*), which is again a *HandoffEvent*. The operations of this type of event is discussed.

[4] Therefore, the user is enqueued only when being admitted and knowing the leave time. The *TimeStamp* of every user in the queue is the leave time of this user.

We now discuss what the system will do to responde to the *LeaveEvent*.

Figure 4.1(D) illustrates the first *LeaveEvent*. At first, the *SystemTimer* is set to the time of this leave event, which is (*N1* + *DN1*). Then the BS releases all resources consumed by this user. Therefore, two metrics *CurrentlyUsedBandwidth* and *CurrentUsersNumber* should be accordingly updated.

Then it is the time to decide the next earliest event again. At this moment, there are three users in the queue whose *TimeStamp* are respectively (*H1* + *DH1*), (*H1* + *H2* + *DH2*) and (*N1* + *N2* + *DN2*). The *SystemTimer* is (*N1* + *DN1*), *HandoffTimer* is (*H1* + *H2* + *H3*) and *NewUserTimer* is (*N1* + *N2* + *N3*). We user the same way mentioned previously to decide the earliest event. First, the earliest *TimeStamp* in the queue is (*H1* + *DH1*). By comparing this *TimeStamp* with two timers, *HandoffTimer* and *NewUserTimer*, we know the earliest time is (*H1* + *DH1*). Because this time is the leave time of a user, this earliest time means the next earliest event is a *LeaveEvent*.

Dynamic Consumed Bandwidth for NRT Connections

The consumed bandwidth of users is varied during the transmission duration in the real world. To simulate this fluctuation feature, the system will let each admitted user change the consumed bandwidth for NRT connections from time to time. In the real world, the change of this kind occurs randomly. If we simulate the change in this way, however, it is hard to model what time the consumed bandwidth changes and how much the change is. Thus, our implementation simply allows all admitted users to change the consumed bandwidth for NRT connections periodically, i.e. every minute. Therefore, at each 1 minute interval, all admitted users in the target cell will be given a chance to change their currently consumed bandwidth for NRT connections.

To any user, the result of the change is random: the consumed bandwidth might be increased or decreased or even remain the same. In order to accurately simulate this variation as much as possible, we need the mean and the standard deviation for a distribution

modelling this variation. This is discussed in Subsection 5.2.1. The simulator lets users specify a coefficient (reflecting the standard deviation) of the variation and then the mean is computed by multiplying the initial bandwidth with the given coefficient parameter. Finally, the simulator takes these two parameters and then generates a series of numbers with a normal distribution, which are the changed bandwidth for each time.

The changed bandwidth has the following two rules: the total consumed bandwidth of all admitted users cannot be more than the bandwidth capacity of the BS at any time and the consumed bandwidth of every user cannot be more than the ($BUB \times$ *Initial Bandwidth*). Acceptance of bandwidth increase depends on the admission control of Section 3.2.1 in Chapter 3.

4.2 Development Environment

The simulation system running the conventional mechanism is named *ThesisBaseCase* and the other simulator running our novel mechanism is called *ThesisMyCase*. Both simulation systems are Windows applications developed in Visual Studio .NET Professional (2002 Version). Because we planned to develop the entire simulation system from scratch and the allowed development time was very tight, we decided to choose the development tool which can bring us the rapid application development (RAD) feature. Visual Studio .NET Professional effectively provides this ability. To cope with this integrated development environment (IDE), the programming languages we employ are Visual C# .NET and Visual C++ .NET. Accordingly, the operating system is Windows XP and the background database system is Microsoft SQL Server 2000. The simulations run on an Intel Pentium III-550 M PC and an Intel Pentium IV-2.6 G PC.[5]

The majority of the code is developed by using Visual C# .NET. Because the underlying runtime environment Microsoft .NET Framework 1.0 does not have predefined control to

[5]Windows, Windows XP, Microsoft SQL Server 2000, Visual Studio .NET, Visual C# .NET and Visual C++. NET are all registered trademarks of Microsoft Corp.. Pentium is the registered trademark of Intel Corp..

let the user to select the folder, we have to use Visual C++.NET to developer a reusable component and convert it to a Dynamic Link Library (DLL) file. Through invoking the low level Win32 Application Programming Interfaces (APIs), this DLL file can provide the functionality of a folder selecter. [6]

The data generated by the simulator should be stored in the disk. People are able to use either plain text or a database to do so. In the latter case, the generated data is first wrapped into Extensible Markup Language (XML) format [31] and then stored into the Microsoft SQL Server 2000 [32].

Both applications have a three-tier architecture: the interface, the simulation logics and the background database. The interface is the way for the user to input the required parameters. The tier of the simulation logics is the core algorithm employed by the simulator to read the parameters, set up the initial status of the system, simulate the events and trigger to corresponding mechanisms, and log the metric data. The tier of the database is responsible for storing data in the disk using either the plain text or the database. Each tier provides some interfaces (i.e. function calls of objects) to communicate with other tier and the internal mechanism of each tier is transparent to other tiers.

4.2.1 Microsoft .NET Framework and C#

Microsoft Visual Studio .NET is based on Microsoft .NET Framework [33] [34], which is a managed, type-safe environment for both application development and execution. The framework is responsible for all aspects of the execution of the program, from allocating the memory to granting access permissions to the application, to managing the entire procedure of application execution, and to eventually releasing the unused memory. The Framework generally includes two critical components: the Common Language Runtime (CLR) [35] [36] and the .NET Framework class library [37].

The CLR essentially is an execution environment for managed code. In the Microsoft

[6] Visual C++ .NET is not necessarily required in this application. However, in order to achieve a degree of the simplicity of user interface, we developer a DLL file using Visual C++ .NET.

terminology, the managed code is the code managed in some environment where the objects are controlled by the garbage collector (GC). For example, in the traditional C++, the object initialized by the operation *new* must be explicitly removed by the operation *delete* or *delete[]*. However, in .NET Framework, releasing the memory of unused objects is now the job of the GC instead of the developer. Thus, the C++ in .NET Framework is called *managed C++* [38]. The CLR provides such services as code compilation, memory allocation, thread management, and garbage collection.

The .NET Framework class library is a collection of architectured and reusable types and interfaces which are designed to integrate with the CLR. These types are object-orientated and the cross-language compatibility, which means multiple languages (up to these days, the number of languages is more than 20, including C/C++, C#, Visual Basic, COBOL, FORTRAN, Perl, etc.) can invoke the same class and same function as long as the application is executed in the .NET Framework. Moreover, the applications developed by different languages can be invoked by each other. Finally, most of the types in .NET Framework class library are fully extensible, which allows developers to build their own types to incorporate customized functionalities into the application. These features dramatically increase the software productivity.

C# (pronounced "c sharp") language [39] was pioneered by Microsoft and submitted to the European Association for Standardizing Information and Communication Systems (ECMA International[7]) to be an international standard *Standard ECMA-334* [40]. After referencing more than 10 previous programming languages, Microsoft introduced this language which embraces Rapid Application Development and emerging Web Programming Standards, brings people strong type-checking and garbage collection, and allows extensive interoperability and versioning.

[7]This organization was founded in 1961 and had the name European Computer Manufacturers Association, i.e. ECMA. It was renamed in 1994 to the current name to reflect the characteristics of its international activities but it took ECMA International as its abbreviation.

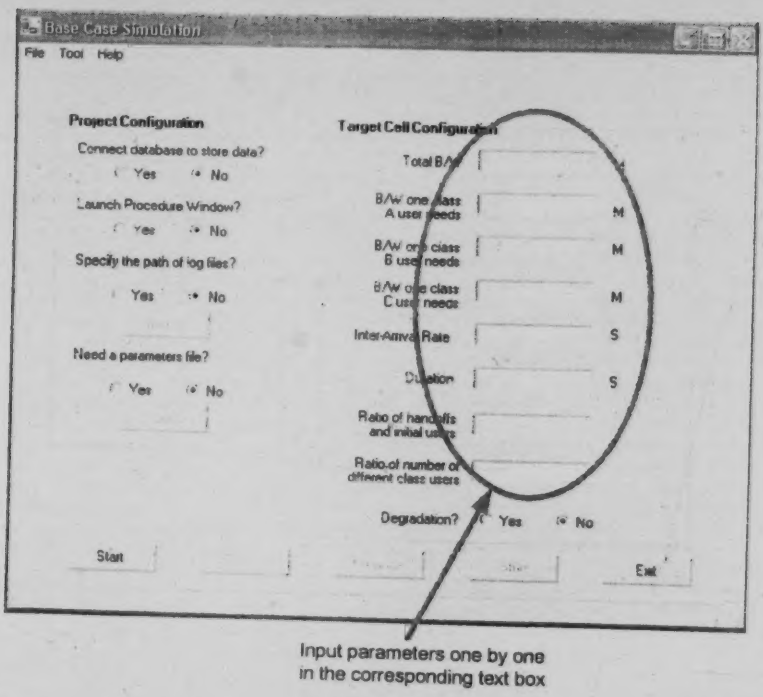

Figure 4.2. THESISBASECASE: USERS CAN INPUT NEEDED PARAMETERS IN TEXT BOXES

4.2.2 The Interface: The Place to Input Parameters

The interface is the place where the user can input the set of parameters needed by the simulator. To facilitate this process, we allow people to either input a number of parameters one by one in the corresponding text box, or allow direct input through parameter files. Figure 4.2 and Figure 4.4 [8] show how to individually input parameters in *ThesisBaseCase* and *ThesisMyCase* respectively. Figure 4.3 and Figure 4.5 illustrate the convenient way to collectively input parameters at a time.

4.2.3 The Simulation Logic

While the CLR loads the application into memory to begin to execute the application, an object *aForm* of the class *System.Windows.Forms.Form* is created to display the user

[8] In both interfaces, there is a checkable option "Launch Procedure Window?". This functionality is not implemented in both applications and will be considered in the future works. Thus, we must select "No" to this option.

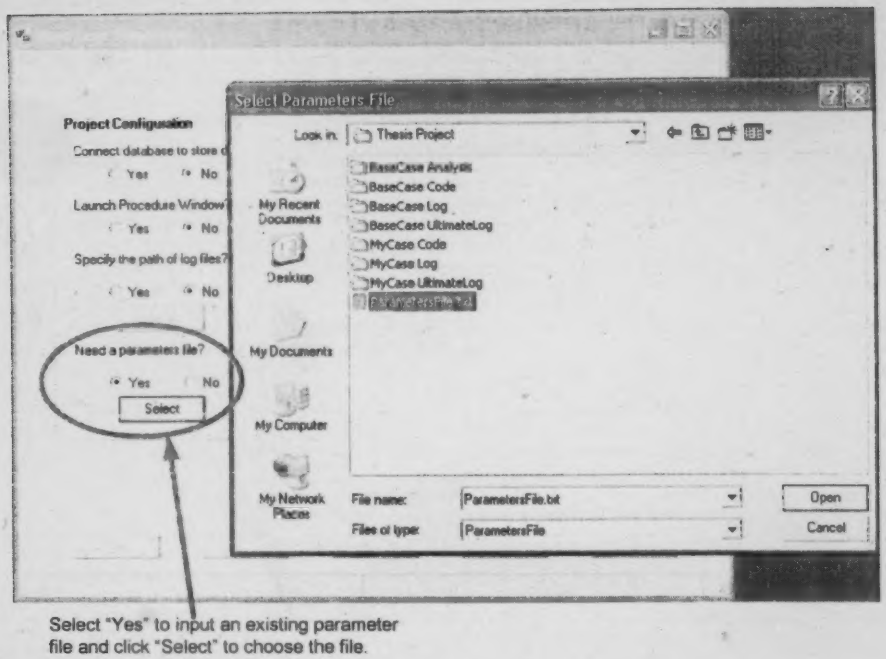

Figure 4.3. THESISBASECASE: USERS CAN INPUT NEEDED PARAMETERS FROM AN EXISTING PARAMETER FILE

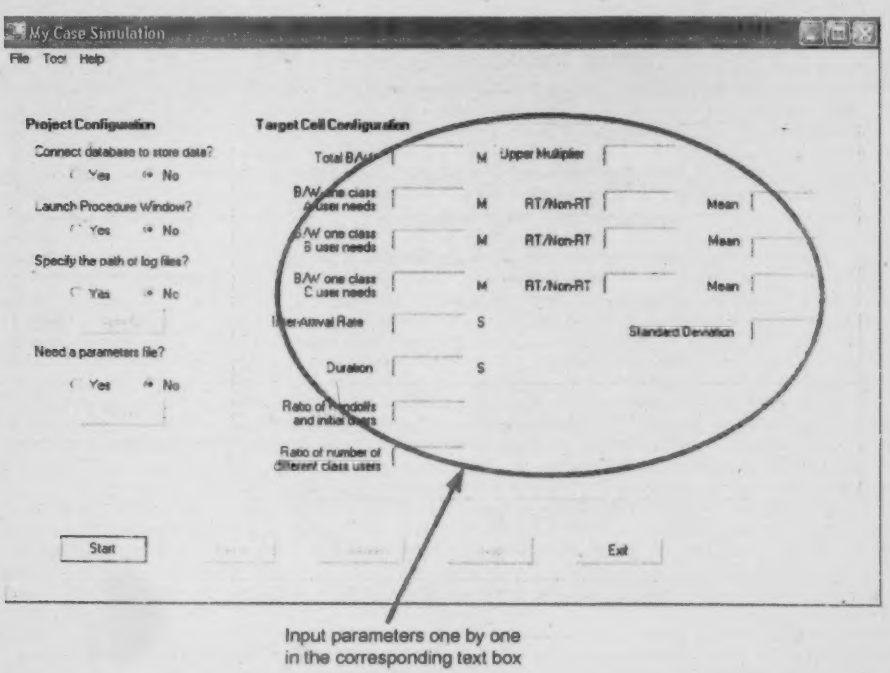

Figure 4.4. THESISMYCASE: USERS CAN INPUT NEEDED PARAMETERS IN TEXT BOXES

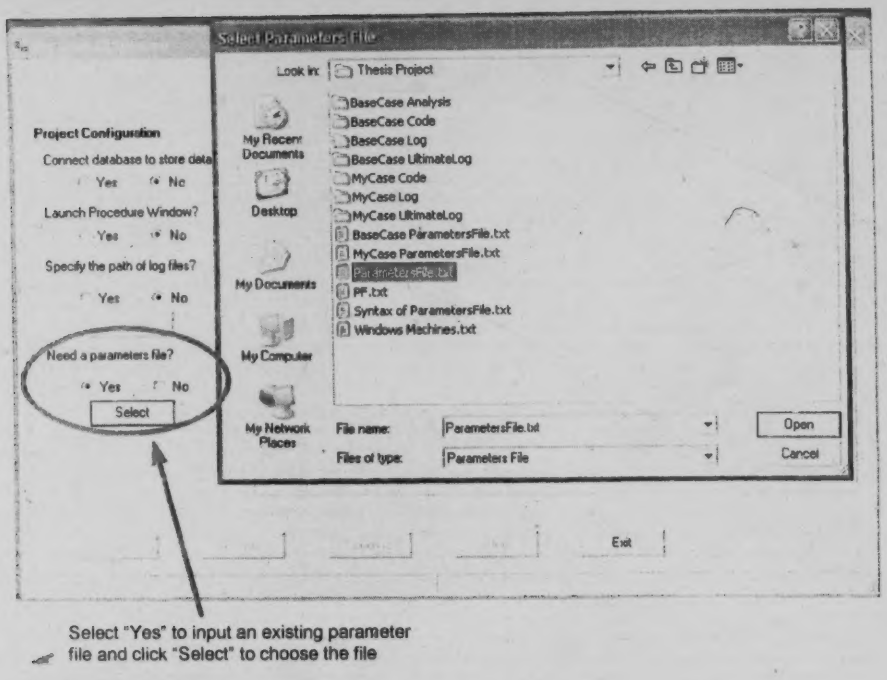

Figure 4.5. THESISMYCASE: USERS CAN INPUT NEEDED PARAMETERS FROM AN EXISTING PARAMETER FILE

interface.

When the use presses the *Start* button after inputting the parameter data or specifying the parameter file, *aForm* initializes a *System.Thread* object *aThread* and a *TargetCell* object *aTargetCell*. The *aThread* object launches a separated thread to run the simulation. The *aTargetCell* simulates a target cell in the real world and it has bandwidth, which can be adjusted by the input parameter. After *aThread* is created, it invokes the *StartSimulate()* method to start the simulation.

4.2.4 Data Storage: Plain Text or Database

Users are allowed to choose either the plain text or the background database to store the data generated in the simulation. In order to efficiently integrate with Visual Studio .NET, the selected background database system is Microsoft SQL Server 2000.

If we decide to store data into plain text, users are asked to specify the destination folder

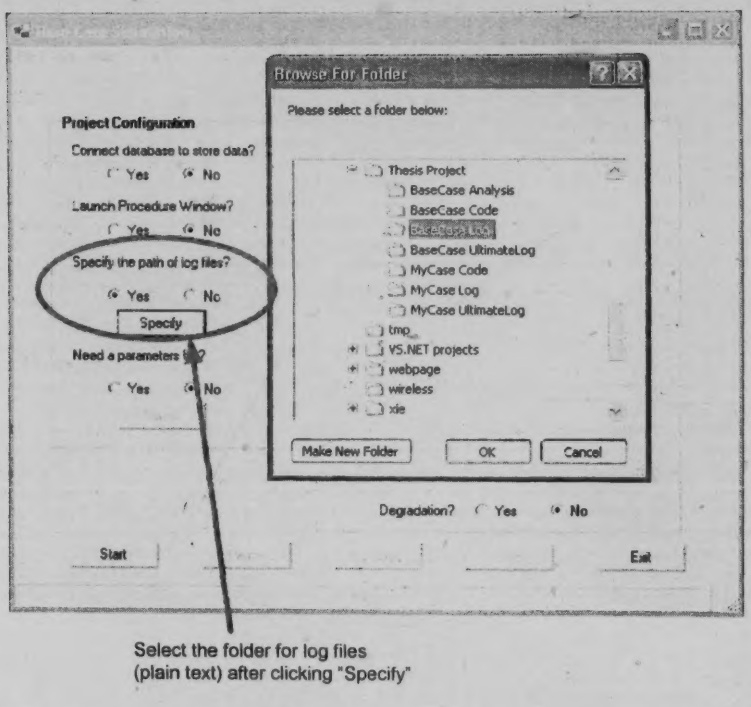

Figure 4.6. THESISBASECASE: USERS CAN SPECIFY THE FOLDER FOR LOG FILES

after clicking *Specify* button and selecting the folder in the resulting *Folder Browser* [9]. After that, the application will save all data into the specified folder. Figure 4.6 and Figure 4.7 show how to specify the folder for log files for *ThesisBaseCase* and *ThesisMyCase* respectively.

If the user prefers to choose the database to store the data, they simply select "Yes" to the first question in the interface *Connect database to store data?*. Then the application will automatically connect the default database system in the current domain and store data. There is no need for users to specify the folder anymore and the button of *Specify* is also disable in this case. In this application, data are first wrapped into XML format and then stored into the database Microsoft SQL Server 2000. Although the current version of the simulator actually does not require XML format, we consider the possible collaboration in the future, in which case the result of our simulator can be read, understood and then used

[9]This folder browser is a DLL file. It is programmed with managed C++ and invoked by C#.

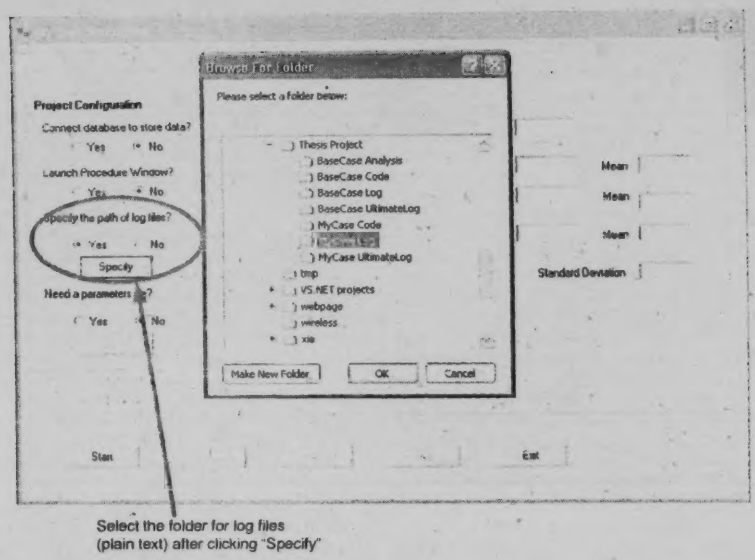

Figure 4.7. THESISMYCASE: USERS CAN SPECIFY THE FOLDER FOR LOG FILES

by different applications in the distributed environments. For example the next version of this simulator could be a web-based simulator and the generated data could be reused by different applications in different platforms through the Internet via the HTTP protocol.

4.3 Verification and Validation

In [30, Chapter 25], Dr. Jain indicates that validation indicates "whether the assumptions are reasonable" and verification indicates "whether the model implements those assumptions correctly". The first term concerns whether or not the assumptions are representative and the second concerns whether or not the system is implemented correctly based on these assumptions.

4.3.1 Verification

In our system development, six verification techniques were employed as follows:

1. Top-Down modular design. Due to the Object-Oriented Programming (OOP) feature of C#/C++, the entire system application is composed with a number of classes

(i.e. modular in this context), each of which simulates some corresponding functions in the real world. For instance, the class `TargetCell` tries to simulate the important functions of the target cell, such as the admission of a new user. The class `User` mimics the characteristics of a general user, including the class and currently consumed bandwidth for RT connections and RNT connections respectively.

2. Antibugging. This means people use additional ways of checking which will report bugs if any. One example is that, if the ratio of the number of the handoff users to the number of new users are parameterized to 1:1, we carefully compare the number of handoff users and the number of new users in the output to see whether they are 1:1. Another example is to check some distributions employed in our applications. We have the application to solely generate a set of random number based on the distribution and then the curve of these numbers is drawn after exporting them to Matlab[10].

3. Self code walk-though. This is actually code debugging line by line by the developer.

4. Trace. This is one of the most useful methods. Thanks to the powerful trace ability of Visual Studio .NET, we line by line run our applications for a long time to see whether or not the change of all metrics and related variables meets our expectation.

5. Graphic displays. Without drawing the graphics, it might be hard to clearly know the tendencies of variables. In the antibugging and trace, we often export some related values to Matlab to draw the curves to tell the tendencies of the curves.

6. Seed independence. We are very careful about this problem so that we select the current machine time as the seed and create different random generators for different random events. For example, the random generator for the arrival events is different from that for the duration time.

[10]Matlab is the trademark registered by Mathwork Corp..

4.3.2 Validation

Validation techniques consists of various aspects. Some techniques used for one simulation system might not be used for another. In our system, three techniques are considered as follows.

1. Assumptions. There are a number of assumptions for our systems. Some of them require setting parameters for the system, like the total bandwidth owned by the cell, in which case we follow the voice of the dominant industry vendor. Some assumptions are for the simplicity of the model, like we assume that the arrival rate and average duration time do not change during the simulation.

2. Input parameter values and distributions. We minimize the difference between the parameter values and real world or industry white papers. The 0.384 Mbps and 0.144 Mbps follow the widely discussed technical white papers in [41]. As to the distributions, we follow the widely used distribution for the same case, like the arrivals follows a Poisson distribution.

3. Output and conclusion. The output of metrics shows that the behavior of curves matches our expectation very well, which indicates that our model and implementation are correct.

4.4 Summary

In this chapter, we mainly discuss the modelling of the simulation, the development of the simulation systems, and how to verify the implementation and validate the model. The objective system is an ordinary target cell. It models the arrival of handoff users and the arrival of new users, whose interarrival time follow exponential distributions. A user stays in the cell a length of time given by an exponential duration. All chosen distributions are based on widely used conventions.

The simulators are developed from scratch. It results in the best customized configuration, which means we only implement what we are concerned with. Our experience shows again that Visual Studio .NET is a rapid Integrated Development Environment, whose powerful debugging functionality facilitates the development process. In consideration of the flexibility of end users, we give end users choices to setup parameters and store the logging data, which is very helpful for large number of sets of parameters and long running simulations.

Verification and validation are mandatory to the simulation. We paid much attention to both of them before we actually started the simulation. Many different techniques are employed to ensure our modelling is reasonable and acceptable, and our implementation is done in a correct way. The final experimental results, discussed in Chapter 5, show that they match our expectation very well. This means our modelling and implementation are successful.

Chapter 5

Simulation Results and Analysis

This chapter concentrates on the performance evaluation and the related analysis. Many other things that are mandatory to the simulation, such as the assumptions and parameters, are discussed as well in this chapter.

Each simulation system has a number of parameters, which are carefully selected and believed to possibly influence the system performance metrics. In addition, each simulation is parameterized, which means people can adjust a set of parameters to do experiments to evaluate the system performance under different situations. The *ThesisBaseCase* has 7 parameters and the *ThesisMyCase* has 16 parameters. Our simulation systems allow end users to setup parameters either by directly inputting them in the interface or by storing required parameters in a separate file `ParametersFile.txt`.

The performance evaluation is the ultimate purpose of the simulation. The simulator generates data of metrics and these data are save into the disk. Afterwards, we compare the data generated from two different simulation systems and evaluate their performance in different experiments, each of which contains a different set of parameters. The comparison of four metrics shows that our novel approach is much better than the ordinary mechanism in most cases.

In this chapter, we first introduce the performance metrics. Then, assumptions and required parameters are discussed in detail. After that, we move to the core of the chapter, the performance evaluation.

5.1 Performance Metrics

First, both simulations must have the same set of performance metrics. Otherwise, it is hard to fairly compare their performance. Among many potential metrics, we carefully chose some important ones as follows.

1. The average bandwidth utilization, which is the totally consumed bandwidth over the capacity of the cell. It is clear that the system wastes the resources if the bandwidth cannot be fully used most of the time. Thus, this metric indicates the efficiency of the resource utilization of the entire system. On the other hand, a system that is easily fully utilized will force the BS to reject users or trigger the bandwidth degradation mechanism. Thus, when evaluating the performance, this metric should be considered with other metrics, like the degree of admission.

2. The average ratio of admitted handoff users, which is the number of the admitted handoff users over the total number of all handoff users. This is an important metric. The rejection of a handoff user leads to a large negative impact on the user experience because the connection of the user is cut off suddenly without any warning while the user is moving. Obviously, the more handoff users admitted, the better the system is.

3. The average ratio of admitted new users, which is the number of the admitted new users over the total number of all new users. Similar to the previous case, this metric indicates the degree of successful initial admission requests.

4. The number of admitted handoff users and the number of admitted new users. Experiments shows that these two metrics have different behavior from the admission ratio of handoff users or the admission ratio of new users. Thus, the analysis of these metrics is an effective supplementary analysis.

5. The number of all admitted users. This denotes the degree of the service provided by the BS. The larger it is, the more people the BS can service at a time. On the other

hand, too many people will easily result in exhausted resources, and then lead to the rejection or to bandwidth degradation. Thus, the larger value of this metric may not mean an advantage and other metrics, like the degree of the rejection, should be considered together.

5.2 Assumptions and Parameters

5.2.1 Assumptions

Assumptions are important to both the simulation modelling and the system implementation. In our simulation system, there are some assumptions as follows:

1. The different classes of users: we assume that the carrier provides services to 3 different classes of users: A, B and C. A is the top class with the highest priority and C is the lowest class with the lowest priority. Each class has different initial bandwidth and BUB which are both user-provided parameters: users of class A have the largest initial bandwidth and BUB, whereas users of class C have the smallest initial bandwidth and BUB.

2. The capacity of the bandwidth of the cell: since the industry standard of 3G is still developing and different vendors issue different white papers, we select the standard [42] employed by the 3G pioneer QUALCOMM Inc.. In this standard, QUALCOMM Inc. considers the capacity of one cell as 7.4 Mbps. Thus, in our simulation experiments, we choose this value as the total bandwidth one cell can own.

3. The bandwidth consumed by users of different classes: the maximum totally consumed bandwidth of a user is 2 Mbps, which corresponds to users of class A. The users of class B are allowed to consume 0.388 Mbps, and users of class C can consume 0.144 Mbps.

4. The distribution of the arrivals: the [30, Chapter 30] mentioned that the commonly used distribution of the arrivals follows the Poisson distribution. This also means that

the interarrival times follow an exponential distribution. In this thesis, we employ the above distribution. However, more detailed distributions could be used instead.

5. The distribution of the duration time: the [30, Chapter 30] indicates that the commonly used distribution of the duration time follows the exponential distribution. In this thesis, we employ the above distribution. However, a more detailed distribution could be used instead.

6. The distribution of the changed bandwidth: at each 1 minute interval, the bandwidth consumed by each user will be changed. The change of this kind is actually random, which can be increased, decreased or remain the same. To model this random change, we assume that in a big enough sampling space, i.e. if the simulation can be run for long enough time, the distribution of the change of this kind follows the normal distribution.

7. The values of BUB and the standard deviation: for simplicity, we assume that in *ThesisMyCase*, the users of different classes have the same predefined BUB and the same standard deviation. *ThesisBaseCase* does not use these two values.

8. Handoff users consuming some resources: at the moment when an incoming handoff user roams into the target cell, we assume that this user is consuming some resources and do not consider the case where this user turns on the cellular phone but does not use any bandwidth[1].

9. The interarrival rate and the average duration time do not change: since the simulation system imitates the target cell working more than a day, these two rates must have some degree of fluctuation during the daytime and at night, i.e. the interarrival rate or the duration time should have a certain degree of difference between the daytime and the night. Regarding these two parameters as constant is not precisely

[1] In fact, it is a possible case in the real world where the cellular phone is turned on but is not in the conversation or is not accessing the Internet at the moment when the user roams across the cells.

accurate. However, we do not know any literature studying the different modelling for two rates in the daytime and the night. We therefore use the constant rate during the entire simulation duration.

5.2.2 Parameters

One feature of our simulation systems is that our simulations are parameterized. Users are allowed to input the set of parameters to initialize the simulator and adjust the characteristics of the simulation. All chosen parameters are supposed to be able to influence the system performance and some of them are shown to have important impact on the performance.

The *ThesisBaseCase* requires 9 parameters; and the *ThesisMyCase* requires them as well and 7 more parameters. We now discuss them respectively.

5.2.3 Parameters Required by ThesisBaseCase

The *ThesisBaseCase* is running the traditional bandwidth allocation mechanism and rejects the user once the resources are exhausted.

1. The total bandwidth the BS has. Because the BS in *ThesisBaseCase* does not have the bandwidth degradation mechanism, this parameter is critical in terms of rejecting users. Intuitively, in the case that the target cell has a constant flood of incoming users and these users do not leave, the more bandwidth the BS has, the later the BS starts to reject users. Thus, this parameter might have great impact on the degree of rejection. The unit of this parameter is megabits.

2. The initial bandwidth for a user of class i where i is A, B or C. As we introduced in Chapter 3, the initial bandwidth is the smallest bandwidth the BS can assign to the user when this user is admitted. Obviously, the more the initial bandwidth is, the sooner the total bandwidth will be exhausted. Because the priority of three classes are different, the initial bandwidth for different class are accordingly different. This

lets us specify three parameters to describe the different initial bandwidth for three different classes. The unit of these parameters is megabits.

3. The average interarrival time. The unit of this parameter is seconds. Because both the incoming handoff users and the new users are arrival events to the target cell, we only specify one interarrival time for both of them and the unit is seconds. However, when generating the random number for the next interval time, the function `HandoffNextExp` and `NewUserNextExp` need the interarrival time for handoff users and new users instead of the interarrival rate of all kinds of users as their arguments (more detailed discussion can be found in Equation 5.1).

4. We specify the parameter *the ratio of the number of handoff users to new users* to compute the interarrival time for handoff users and new users. This process is described by the following equation.

   Definition 3 *Let the arrival rate of all users be R. Let the ratio of the number of handoff users to new users be a:b. Let the arrival rates of handoff users and new users be x and y respectively. Thus, x and y can be computed as follows:*

   $$\begin{cases} x &= \frac{a}{a+b}R \\ y &= \frac{b}{a+b}R \end{cases} \tag{5.1}$$

   Another choice of this parameter is that we specify two parameters for interarrival rates for handoff users and new users respectively, and do not need the parameter *the ratio of the number of handoff users to new users*. We finally did not select this choice in consideration of some different characteristics between the handoff users and the new users. The difference of this kind and the proportion of the number of users of two types *might* affect the performance. Thus, we need the ratio of number of two types of users.

5. The average duration time of the mobile users staying in the cell. The unit of this parameter is seconds.

6. The ratio of the number of users of class A, B and C respectively. Since users of different classes consume different initial bandwidth and enjoy different priorities, intuitively, the proportion of the number of users of three classes *might* influence the system performance.

5.2.4 Parameters Required by ThesisMyCase

ThesisMyCase is running our novel bandwidth allocation mechanism and will launch the bandwidth degradation mechanism when the resources are exhausted. All parameters required by *ThesisBaseCase* are the same as some needed by *ThesisMyCase*. We explained those that are different only. Some further parameters are required as follows:

1. The value of BUB. As the multiplication coefficient, it means the degree of the freedom of asking for more resources. It is only used in *ThesisMyCase*. More details can be found in the definition of BUB (Equation 3.3) in Chapter 3.

2. The ratios of the consumed bandwidth for RT connections to NRT connections (for three different classes). In *ThesisMyCase* which has the notion of RT and NRT, different connections have their own characteristics. Thus, the proportion of each connections *might* impact the performance.

3. Two multiplication coefficients. The bandwidth consumed by NRT connections is dynamic and will be changed by the simulator for every 1 minute. This change is uniformly random but the variance *might* influence the system performance. Thus, we allow the user to control the variance via parameters to see whether it does has the influence. First, the simulator takes one coefficient to compute the mean bandwidth consumed by users of three different classes by multiplying the initial bandwidth with the given coefficient. For example, coefficient $\times$ initial bandwidth of class A is the mean bandwidth consumed by users of class A. In order to have a fair comparison with conventional case, we give the coefficient for the mean bandwidth the value 1.0

in the *ThesisMyCase*. Then the simulator uses the other coefficient and this computed mean bandwidth to calculate the standard deviation of total bandwidth consumed by one user. That is, the simulator multiplies the mean bandwidth with the second coefficient to get the standard deviation. For example, suppose the mean bandwidth of users of class A is 1.8 Mbps and the given coefficient is 0.1, the standard deviation of all bandwidth consumed by users of class A is ($1.8 \times 0.1 = 0.18$). Intuitively, very large or very small standard deviation *might* influence the performance differently.

5.2.5 The format of ParamtersFile.txt

The parameter file must be named `ParametersFile.txt`. Although the parameter file is the plain text file, the parameter files for *ThesisBaseCase* and *ThesisMyCase* have different format, since two simulators have different parameters.

The typical example of the `ParametersFile.txt` for *ThesisBaseCase* is shown as follows [2].

```
------------------      // must have this line to separate cases
900  //total bandwidth owned by the BS
3    //initial bandwidth assigned to users of class A
2    //initial bandwidth assigned to users of class B
1    //initial bandwidth assigned to users of class C
1    //interarrival rate (unit:  second)
1500 //duration time in the target cell (unit:  second)
1:1.0//ratio of the number of handoff users to the number of new users
1:2:3//ratio of the number of users of three classes A, B and C
false//system does not have degradation mechanism
```

The first line of `ParametersFile.txt` must be a line of minus symbols. Due to the use of parameter file, users can input a number of sets of parameters in one file and these sets of parameters are separated by a line of minus symbols. The following example illustrates the look of `ParametersFile.txt` with multiple sets of parameters. Then the simulator will

[2] Double slashes and the comments are only to explain the meaning of the format. `ParametersFile.txt` actually does not contain the double slashes and the comments.

automatically execute the experiments with different sets of parameters one by one. This approach greatly facilitates the operation of users.

```
------------------
    // the first set of parameters here
------------------
    // the second set of parameters here
------------------
    // the third set of parameters here
```

Because the parameters needed in *ThesisMyCase* are different from those used in *ThesisBaseCase*, the corresponding `ParametersFile.txt` is accordingly different. The typical example of the `ParametersFile.txt` for *ThesisMyCase* is shown as follows [3].

[3]Double slashes, the comments and the blank lines in comments are only to explain the meaning of the format. `ParametersFile.txt` actually does not contain the double slashes and the comments.

```
------------------      // must have this line to separate cases
900  // the total bandwidth owned by the BS
3    // the value of BUB
3    //initial bandwidth assigned to users of class A
1:1.0//ratio of consumed bandwidth for RT to NRT connections
               (users of class A)
0.9  //an coefficient to indicate the mean of consumed bandwidth
               of users of class A
2    //initial bandwidth assigned to users of class B
1:1.0//ratio of consumed bandwidth for RT to NRT connections
               (users of class B)
0.9  //an coefficient to indicate the mean of consumed bandwidth
               of users of class B
1    //initial bandwidth assigned to users of class C
1:1.0//ratio of consumed bandwidth for RT to NRT connections
               (users of class C)
1.0  //an coefficient to indicate the mean of consumed bandwidth
               of users of class C
0.1  //standard deviation (used for the oscillation of consumed
               bandwidth)
10   //inter-arrival rate (unit:  second)
1500 //duration time for a user after being admitted (unit:  second)
1:1.0//ratio of the number of handoff users to the number of new users
1:2:3//ratio of the number of users of three classes A, B and C
```

The line of minus symbols plays the same role as we introduce above for *ThesisBaseCase*. In practical, `ParametersFile.txt` for *ThesisMyCase* also has a number of sets of parameters separated by the line of minus symbols.

5.3 Performance Evaluation

This section is the core of this chapter. Since the simulation systems are parameterized, we run the simulation with different sets of parameters. For reasons discussed in Section 5.2.1 *Assumptions*, some parameters are fixed and the rest are changed to observe the influence of these parameters. To minimize the influence of random numbers, we remove the transient period of performance curves, plotted by Matlab. Eventually, we analyze the performance curves of two simulation systems to draw some conclusions.

5.3.1 Experiments

To minimize the variance brought by the random events, we repeat the computation with the same set of given parameters for 8 times. For each run, the simulator runs for 8000 log times[4] where the simulator logs the corresponding metrics at 10 minutes interval (on the basis of the simulation time instead of the real time people use). Therefore, this is equivalent to a simulation length of 55.55 days. The simulator averages results for 8 of these runs. The following pseudo-code illustrates this procedure.

```
for(int repeat = 0; repeat < 8; repeat++){
     for(int logCount = 0; logCount < 8000; logCount++) {
        //run the simulation and log for every 10 minutes
     }
}
```

Based on the assumptions, the total bandwidth we decide for the target cell is 7.4 Mbps. We consider interarrival times of 2, 3, 5, 10, and 30 respectively, and duration times of 60, 180, 300, 420, 600, 780, 900, 1020, 1200, 1500, and 1800 respectively. The ratio of the number of users of three different classes is 1:2:3. The ratio of the number of handoff users to the number of new users is 1:0.5, 1:1, and 1:2. For the novel approach, the parameter of the standard deviation is 0.9.

5.3.2 Transient Removal

Most simulations have a transient period before the performance curve becomes steady. This rule holds in our systems as well. This transient period must be removed before the analysis of the performance; otherwise, the final analysis may be influenced. Two methods are used to remove the transient period: long run and truncation.

[4]In fact, we test the simulator to decide this number. We found in our testing cases, the performance curve becomes stable after logging for 300 times or 400 times. To maximize the accuracy and the sampling space, however, we decide to run the simulator for 8000 times.

1. Long run. Obviously, a long enough execution will minimize the negative influence of the transient period. To decide the execution duration, we run the system by setting different number of logs (this variable directly decides the duration of execution) and then draw the performance curve versus time in Matlab to see the change of the curves. First, we give an extremely big value, like 10000, and draw the curve; and then we give a extremely small value, like 1000, and draw the curve again. By using this binary-search-like method, we find the curve will be steady after the first several hundreds of logging. However, to minimize the negative influence of the random number, we finally decide the number of logs to be 8000.

2. Truncation. When drawing the overall curves, we easily observe that the curves of our systems in the transient period is not steady. We ignore the values of the first several hundreds (normally 300 - 400) and only consider the rest of the values. Because the execution duration (8000) is much larger than the transient period (300 - 400), the influence of the average change is very small.

5.3.3 Performance Comparison

Average Admission Ratio

Figure 5.1 illustrates the admission ratio of handoff users versus the duration time with different interarrival times. The ratio of the number of users of three different classes is 1:2:3 and the ratio of number of handoff users to the number of new users is 1:1. In this figure, the dashed lines are the conventional approach and the solid lines are our approach; the numbers close to the curves are the corresponding interarrival times.

It is obvious that the novel approach is better than the corresponding conventional approach in all cases. In some cases our approach even beats the conventional one by 60%, such as at low load, when the average interarrival time is 30 and the average duration time is 1800 and when the interarrival time is 10 and the duration time is 600. Our approach can admit almost 100% of the handoff users when the interarrival time is 30 and the duration

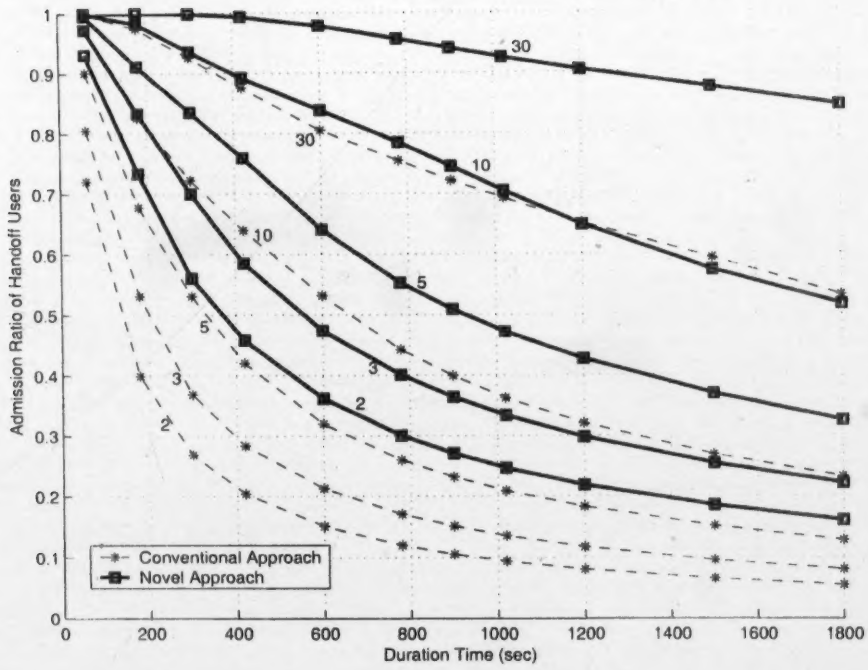

Figure 5.1. COMPARISON OF THE AVERAGE ADMISSION RATIO OF HANDOFF USERS

time is less than 420.

In addition, both systems have the same tendencies as follows. We let R_{AH} be the admission ratio of handoff users, let T_s be the duration time, and let R_{ia} denote the average interarrival time.

1. When the average interarrival time increases, the admission ratio of handoff users increases, i.e. this metric is directly proportional to the interarrival time as follows:

$$R_{AH} \propto R_{ia} \tag{5.2}$$

The longer interarrival time means that the users come less frequently, which accordingly lets resources be not easily exhausted. Therefore, it is easier for the BS to admit the handoff users. As the figure shows, we can observe the trend that the longer the interarrival time is, the closer both solid lines and dashed lines are to 1.0.

2. When the duration time increases, the admission ratio of handoff users decreases, i.e.

the admission ratio is inversely proportional to the duration time as follows:

$$R_{AH} \propto 1/T_s \tag{5.3}$$

The longer duration time means that the user holds the resources for a longer time, which will more easily result in exhausted resources. As the figure shows, the longer the duration time is, the further both solid lines and dashed lines are away from 1.0.

3. When the duration time increases, the admission ratio of handoff users decreases more sharply for small duration times than for large duration times, i.e. the absolute value of the first order partial derivative of the curve is inversely proportional to the duration time as follows:

$$\left|\frac{\partial R_{AH}}{\partial T_s}\right| \propto 1/T_s \tag{5.4}$$

As the figure shows, the curve is sharper for the smaller duration time than for the larger duration time. For example, when the interarrival time is 2 in the conventional approach, the curve decreases more than 30% while the duration time increases from 60 to 180, whereas the same curve drops less than 10% after the duration time is 300 and larger. The same thing happens to the novel approach. We believe this is because that when the duration time increases, the system is close to the saturated status since the users stay longer in the cell. Hence, the curve of the admission ratio gradually becomes smooth.

4. When the duration time increases, the admission ratio of handoff users decreases more sharply for small interarrival time than for large interarrival time, i.e. the absolute value of the first order partial derivative of the curve is inversely proportional to the interarrival time as follows:

$$\left|\frac{\partial R_{AH}}{\partial T_s}\right| \propto 1/R_{ia} \tag{5.5}$$

As the figure shows, the curve is sharper for the smaller interarrival time than that for the larger interarrival time. For instance, in the conventional approach, when

the interarrival time is 2, the curve drops more than 30% while the duration time increases from 60 to 180. Whereas if the interarrival time is 30, it decreases only around 20% when the duration time increases from 60 to 180. The same trend can be observed in the novel approach. The possible reason is that when the interarrival time is large enough, the change of the interarrival time does not have significant impact on the performance.

Average Number of Admitted Handoff Users

Figure 5.2 illustrates the number of admitted handoff users versus the duration time with different interarrival time. The ratio of the number of users of three different classes is 1:2:3 and the ratio of number of handoff users to the number of new users is 1:1. In this figure, the dashed lines are conventional approach and the solid lines are our approach; the numbers close to the curves are corresponding interarrival time.

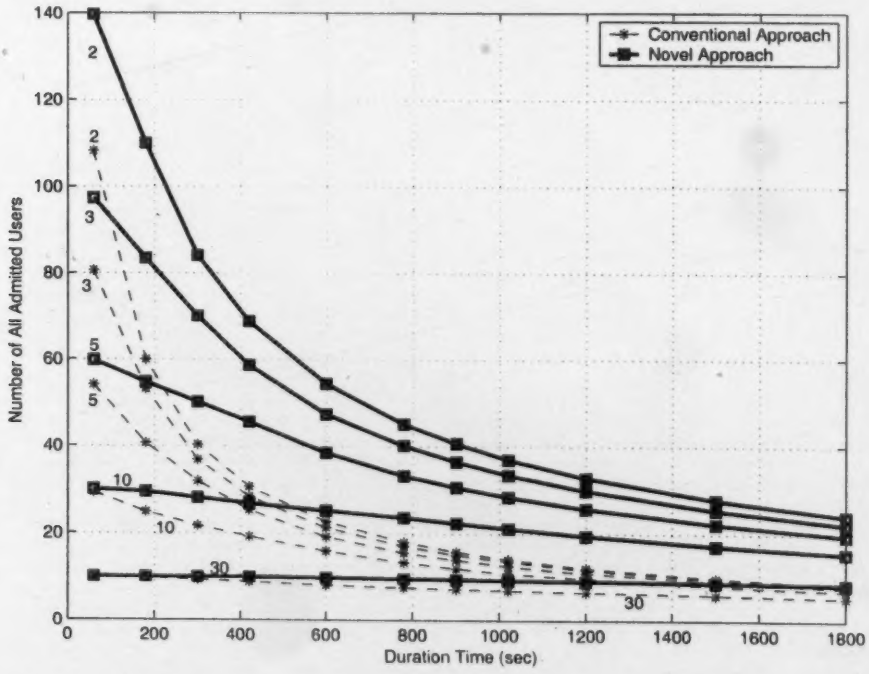

Figure 5.2. COMPARISON OF THE AVERAGE NUMBER OF ADMITTED HANDOFF USERS

In addition, both systems have the same tendencies as follows. We let R_{AHnum} be the

number of admitted handoff users, let T_s be the duration time, and let R_{ia} denote the interarrival time. We mainly discuss one trend and list the rest because one trend is opposite to that observed in Figure 5.1 and the rest are the same.

1. When the interarrival time increases, the number of admitted handoff users decreases, i.e. this metric is inversely proportional to the interarrival time as follows:

$$R_{AHnum} \propto 1/R_{ia} \tag{5.6}$$

The longer interarrival time means that the handoff users come less frequently and fewer handoff users will be admitted. As the figure shows, we can observe the trend that the longer the interarrival time is, the closer both solid lines and dashed lines are to x-axis.

Comparing Equation 5.6 and Equation 5.2, we observe that when the interarrival time increases, the number of admitted handoff users decreases whereas the admission ratio increases. That means in this case, fewer handoff users enter the target cell and they are more easily admitted. This is because the larger interarrival time means users come less frequently and it is relatively easier for the BS to admit a few users instead of the flood of users. For example, given a period of time. If the interarrival time is large enough, there are only a few users arriving, but the BS may be able to admit all of them. If the interarrival time is small enough, hundreds of users might send admission requests and the BS may only admit some of them, say 50%. Therefore, the admission ratio in the first case (100%) is better than in the second case (50%), though the number of users in the first case (several users) is less than the second case (hundreds of users).

2. Since the rest of the trends are the same as those observed in Figure 5.1, we only briefly list them here.

   (a) The number of admitted handoff users is inversely proportional to the duration time.

(b) The absolute value of the first order partial derivative of the curve is inversely proportional to the duration time.

(c) The absolute value of the first order partial derivative of the curve is inversely proportional to the interarrival time

Average Admission Ratio of New Users

Figure 5.3 illustrates the admission ratio of new users versus the duration time with different interarrival time. The ratio of the number of users of three different classes is 1:2:3 and the ratio of number of handoff users to the number of new users is 1:1. In this figure, the dashed lines are conventional approach and the solid lines are our approach; the numbers close to the curves are corresponding interarrival time.

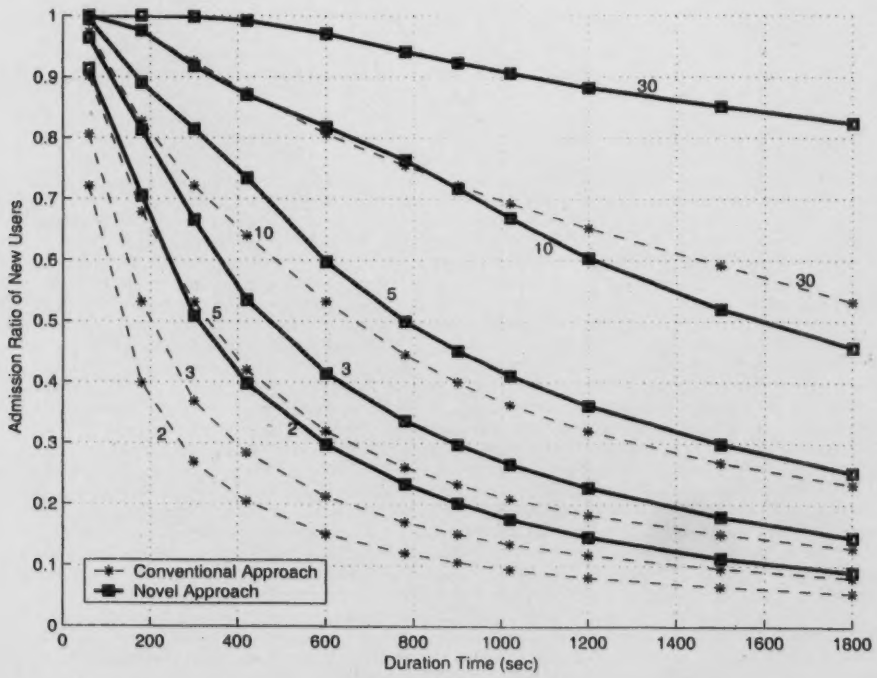

Figure 5.3. COMPARISON OF THE AVERAGE ADMISSION RATIO OF NEW USERS

The characteristics of all curves in this figure are the same as those observed in Figure 5.1 because the ratio of the number of handoff users to the number of new users is 1:1

in these experiments. It is obvious that the novel approach is better than the conventional approach in all cases. In some cases our approach even beats the conventional one by 30%, such as when the interarrival time is 30 and the duration time is 1800 and when the interarrival time is 10 and the duration time is 780. The novel approach even can admit almost 100% handoff users when the interarrival rate is 30 and the duration time is less than 420.

The trends shown in this figure are the same as those shown in Figure 5.1. We let R_{AN} be the admission ratio of new users, let T_s be the duration time, and let R_{ia} denote the interarrival time.

1. The admission ratio of new users increases is directly proportional to the interarrival time:

$$R_{AN} \propto R_{ia} \tag{5.7}$$

2. The admission ratio of new users decreases is inversely proportional to the duration time:

$$R_{AN} \propto 1/T_s \tag{5.8}$$

3. The absolute value of the first order partial derivative of the curve is inversely proportional to the duration time:

$$\left|\frac{\partial R_{AN}}{\partial T_s}\right| \propto 1/T_s \tag{5.9}$$

4. The absolute value of the first order partial derivative of the curve is inversely proportional to the interarrival time.

$$\left|\frac{\partial R_{AN}}{\partial T_s}\right| \propto 1/R_{ia} \tag{5.10}$$

Average Bandwidth Utilization

Figure 5.4 illustrates the bandwidth utilization versus the duration time with different interarrival times. The ratio of the number of users of three different classes is 1:2:3 and the

ratio of number of handoff users to the number of new users is 1:1. In this figure, the dashed lines are the conventional approach and the solid lines are our approach; the numbers close to the curves are corresponding interarrival time.

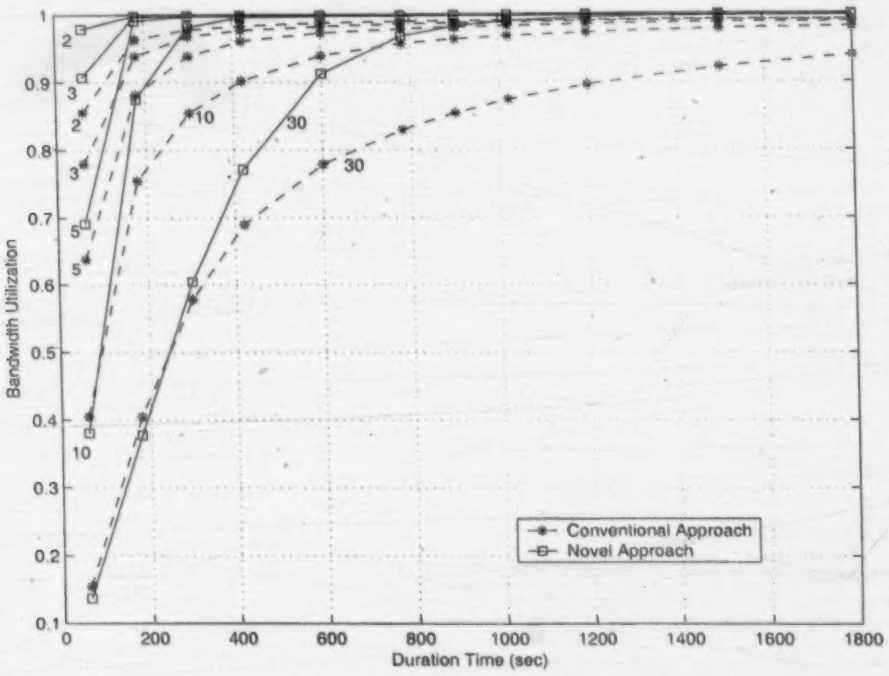

Figure 5.4. COMPARISON OF THE AVERAGE BANDWIDTH UTILIZATION

It is obvious that the novel approach is better than the conventional approach in most cases except two exceptional cases where the difference between our approach and the conventional approach is not significant. In some cases our approach beats the conventional one by 15%, such as when the interarrival rate is 30 and the duration time is 780. For the novel approach, they can use almost 100% resources sooner or later. However, all conventional cases can only be very close to 100% but still lower than the novel approach.

As to two exceptional cases, the first one occurs when the interarrival time is 10 (the second largest value of this variable ranging from 2 to 30) and the duration time is from 60 (the smallest value of this variable ranging from 60 to 1800) to a certain number between 60 to 180. The second exceptional case is observed when the interarrival time is 30 (the

largest value of this variable from 2 to 30) and the duration time is from 60 (the smallest value of this variable ranging from 60 to 1800) to a certain number between 60 to 300. In these two exceptional cases, the duration time is the smallest value and the interarrival time is the largest and the second largest values. This means that very fewer users come up (one user coming up every 10 seconds and 30 seconds) and they stay in the cell for fairly short period of time (slightly more than 1 minutes). In this case, since the system does not have to handle many users staying very long time, the system is pretty idle. Figure 5.4 shows that in these two cases, the system is far from busy and its utilization varies from 5% to 50%. When the utilization is more than 50%, the outstanding performance of our approach is gradually shown. Therefore, we believe the possible reason for these two exceptional cases is that the our approach will show its outstanding performance in busier system where there are heavier workloads, whereas it does not show much difference from the conventional approach in idler system where there are slighter workloads.

In addition, both systems have the same tendencies as follows. We let U_{BW} be the bandwidth utilization, let T_s be the duration time, and let R_{ia} denote the interarrival time.

1. When the interarrival time increases, the bandwidth utilization decreases, i.e. this metric is inversely proportional to the interarrival time as follows:

$$U_{BW} \propto 1/R_{ia} \tag{5.11}$$

   The larger interarrival time means that the arrival users come less frequently, which subsequently means that less bandwidth will be consumed to meet requests of users. Thus, the total bandwidth will not be easily fully utilized. As the figure shows, when the interarrival time increases, both solid lines and dashed lines are further away from 1.0.

2. When the duration time increases, the bandwidth utilization increases, i.e. this metric is directly proportional to the duration time as follows:

$$U_{BW} \propto T_s \tag{5.12}$$

The larger duration time means that the assigned bandwidth will be hooked for longer time. Thus, to deal with the other incoming users, more available bandwidth will be assigned for users. Thus, it is easier for the BS to assign more and more bandwidth to users. As a result, the bandwidth utilization increases. As the figure shows, when the duration time increases, both solid lines and dashed lines are gradually closer to 1.0.

3. When the duration time increases, the bandwidth utilization increases more sharply for small duration times than for larger duration time, i.e. the first order partial derivative of the curve is inversely proportional to the duration time as follows:

$$\frac{\partial U_{BW}}{\partial T_s} \propto 1/T_s \tag{5.13}$$

As the figure shows, the curve is sharper for the smaller duration time than for the larger duration time and the curve is even like a horizontal line when the duration time is large enough. For example, when the interarrival time is 30 in the conventional approach, the curve increases more than 20% while the duration time increases from 60 to 180, whereas the same curve increases less than 10% after the duration time is 300 and larger. The same rule happens to the curve of the novel approach. When the interarrival time is 30, the curve increases more than 20% while the duration time increases from 60 to 180, whereas the same curves only increases less than 10% after the duration time is 600 and larger. We believe the possible reason is when the duration time is large, the system is close to saturated status since users stay in the cell for longer time. Thus, we observe that the curve of the utilization is close to 100% and move smoothly.

4. When the duration time increases, the bandwidth utilization increases slowly for small interarrival times and moves greatly for large interarrival times, i.e. the first order partial derivative of the curve is directly proportional to the interarrival rate.

$$\frac{\partial U_{BW}}{\partial T_s} \propto R_{ia} \tag{5.14}$$

As the figure shows, the curve is sharper for the larger interarrival time than for the smaller interarrival time. For instance, in the conventional approach, when the interarrival time is 2, the curve increases by 10% while the duration time increases from 60 to 180. Whereas if the interarrival time is 30, it increases more than 20% when the duration time increases from 60 to 180. The same trend can be observed in the novel approach. The possible reason is when the interarrival time is large enough, it has relatively smaller impact on the performance.

Average Number of All Admitted Users

Figure 5.5 illustrates the number of all admitted users in the target cell versus the duration time with different interarrival time. The ratio of the number of users of three different classes is 1:2:3 and the ratio of number of handoff users to the number of new users is 1:1. In this figure, the dashed lines are the conventional approach and the solid lines are our approach; the numbers close to the curves are corresponding interarrival time.

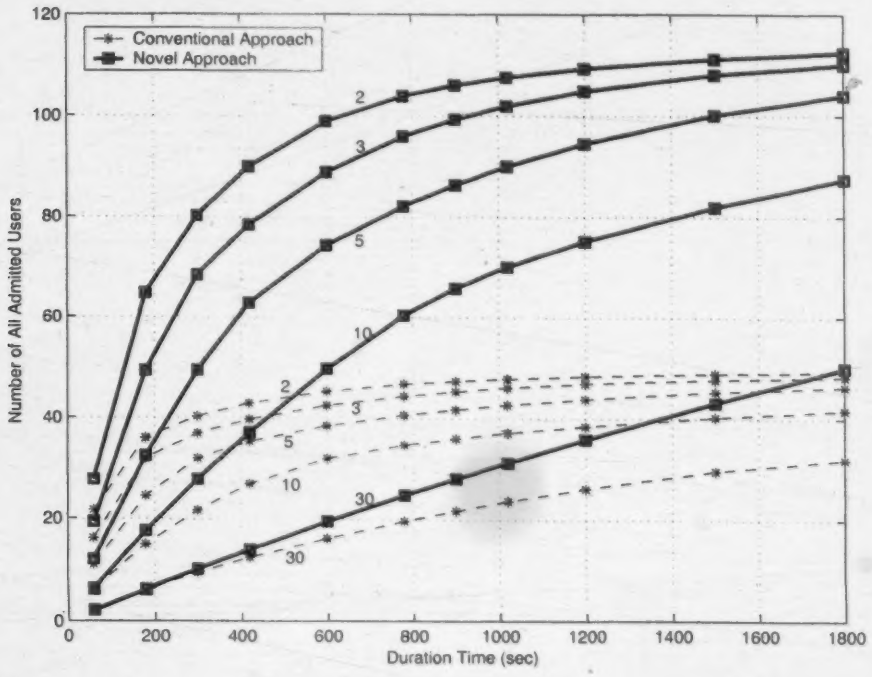

Figure 5.5. COMPARISON OF THE AVERAGE NUMBER OF ALL ADMITTED USERS

It is clear that the novel approach is better than the conventional approach in all cases. In some cases our approach is two times more than the conventional one, such as when the interarrival time is 2 and the duration time is 600 and when the interarrival time is 5 and the duration time is 1500.

In addition, both systems have the same tendencies as follows. We let N be the number of all admitted users, let T_s be the duration time, and let R_{ia} denote the interarrival time.

1. When the interarrival time increases, the total number of all admitted users decreases, i.e. this metric is inversely proportional to the interarrival time as follows:

$$N \propto 1/R_{ia} \tag{5.15}$$

   The larger interarrival time means that the arrival new users will come less frequently, which subsequently means that the BS will handle less users. Thus, the total number of users will be less. As the arrow in the figure shows, when the interarrival time increases, both solid lines and dashed lines are closer to x-axis.

2. When the duration time increases, this metric increases, i.e. it is directly proportional to the duration time as follows:

$$N \propto T_s \tag{5.16}$$

   The larger duration time means that the admitted user will stay at the target cell for longer time. Straightforwardly, there will be more users in the target cell. As the curve shows, when the duration time increases, both solid lines and dashed lines increases.

3. When the duration time increases, the number of users increases greatly for small duration time and moves slowly in large duration time, i.e. the first order partial derivative of the curve is inversely proportional to the duration time as follows:

$$\frac{\partial N}{\partial T_s} \propto 1/T_s \tag{5.17}$$

As the figure shows, the curve is sharper for the smaller duration time than for the larger duration time. For example, when the interarrival time is 2 in the conventional approach, the curve increases around 15 while the duration time increases from 60 to 180, whereas the same curve almost keeps the same value after the duration time is 1000 and larger. The same thing happens to the curve of the novel approach. When the interarrival time is 2, the curve increases almost 40 while the duration time increases from 60 to 180, whereas the same curves only increases slightly after the duration time is 780 and larger.

4. When the duration time increases, the total number of users increases greatly for small interarrival time then moves slowly in large interarrival time, i.e. the first order partial derivative of the curve is inversely proportional to the interarrival time.

$$\frac{\partial N}{\partial T_s} \propto 1/R_{ia} \tag{5.18}$$

As the figure shows, the curve is sharper for the smaller interarrival time than that in the larger interarrival time. For instance, in the conventional approach, when the interarrival time is 2, the curve increases around 15 while the duration time increases from 60 to 180. Whereas if the interarrival time is 30, it increases only around 5% when the duration time increases from 60 to 180. The same trend can be observed in the novel approach.

Influence of Different Ratio of Number of Handoff Users to Number of New Users

To observe the influence of different ratios of number of handoff users to number of new users, we experiment with different values of this metric, including 1:0.1, 1:0.5, 1:1, and 1:2 where the interarrival time is 2, 5 and 30. The following series of figures present the results of experiments. Due to many figures, we select some of them to discuss the general trend. In all figures, the dashed lines are conventional approach and the solid lines are our approach.

1. Figure 5.6 presents the comparison of average admission ratio of handoff users versus the duration time with varied ratio of number of handoff users to number of new users. In this figure, the interarrival time is 2.

2. Figure 5.7 presents the comparison of average admission ratio of new users versus the duration time with varied ratio of number of handoff users to number of new users. In this figure, the interarrival time is 5.

3. Figure 5.8 presents the comparison of average bandwidth utilization versus the duration time with varied ratio of number of handoff users to number of new users. In this figure, the interarrival time is 30.

4. Figure 5.9 presents the comparison of average number of all admitted users versus the duration time with varied ratio of number of handoff users to number of new users. In this figure, the interarrival time is 2.

Observation shows that the conventional approach is not affected by this varied ratio while the ratio has subtle impact on our approach. It is obvious that this varied ratio does not influence the performance curve of the conventional approach because those curves perfectly overlap together. This can be explained that our implementation of the conventional approach does not favor handoff users and both handoff users and new users are admitted in the same way. To our approach, the varied ratio slightly influences the performance curve. When the duration time is small enough, curves of our approach almost overlap together. While the duration time is larger, they begin to show the difference. A slight trend is that the larger ratio of number of handoff users to number of new users will lead to a slightly larger average admission ratio. This indicates that the handoff users have slightly higher priority than new users in our approach and this priority is shown when the number of handoff users is more outstanding, though this trend may not be significant.

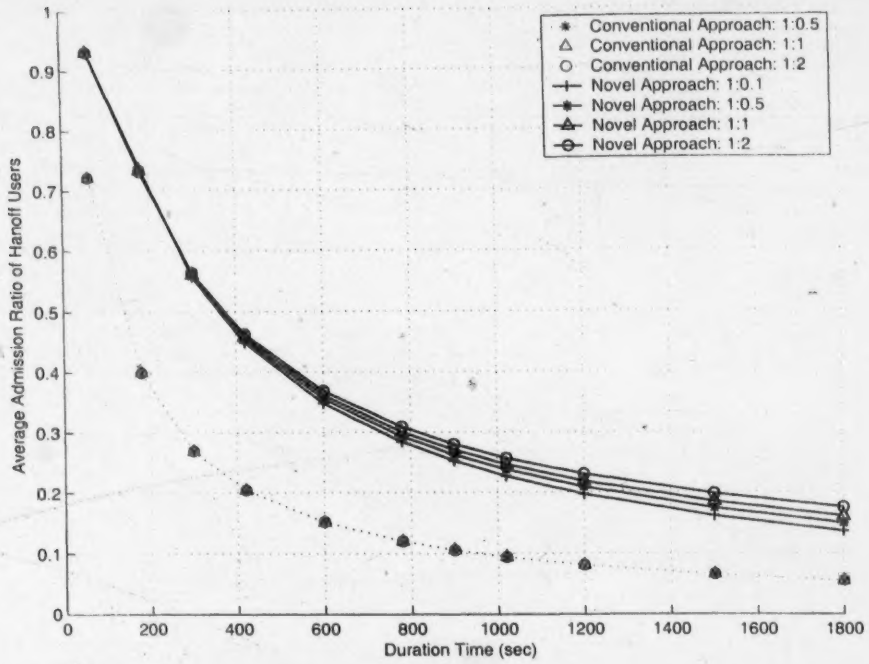

Figure 5.6. COMPARISON OF AVERAGE ADMISSION RATIO OF HANDOFF USERS WITH DIFFERENT RATIO OF TWO TYPES OF USERS

Admission Ratio of Handoff Users vs. Admission Ratio of New Users

Figure 5.10 presents the comparison between the average admission ratio of handoff users and the average admission ratio of new users (versus the interarrival time). The dashed lines are conventional approach and the solid lines are our approach. In this figure, the interarrival time is 2, 3, 5, 10 and 30 respectively; the duration time is 60, 780 and 1800, which are presented by the numbers close to curves.

Something we can observe from this figure is the same as those observed from Figure 5.1, such as how the admission ratio changes when the interarrival time or the duration time changes. In addition, this figure shows that in our approach, the admission ratio of handoff users is larger than that of new users, though this conclusion is not significant when the duration time is 60. This is because we believe that the failure of the handoff has worse impact on the user experience than the failure of admission request from a new user. Thus,

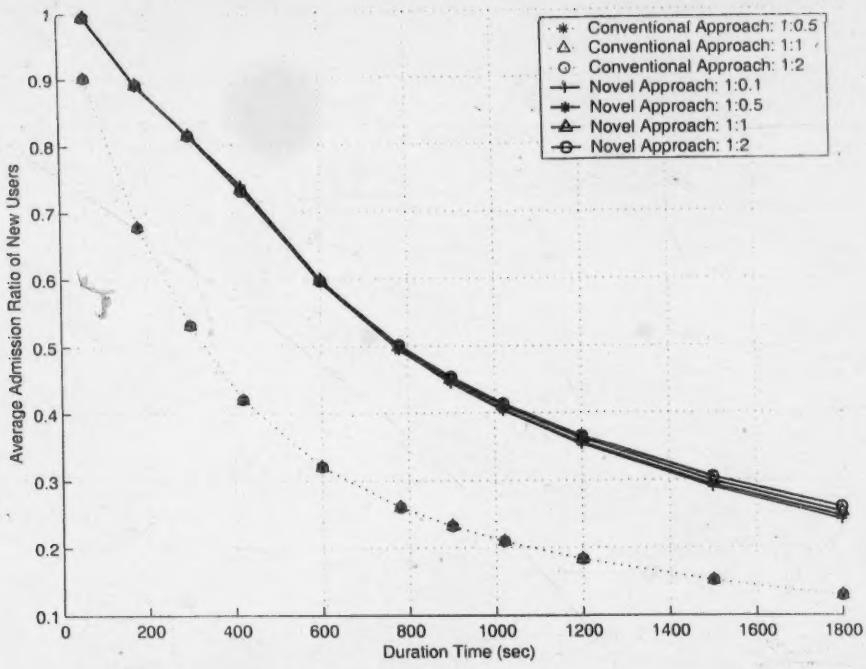

Figure 5.7. COMPARISON OF AVERAGE ADMISSION RATIO OF NEW USERS WITH DIFFERENT RATIO OF TWO TYPES OF USERS

to achieve good user experience, our approach favors the handoff users over new users. This also verifies our implementation. However, in the conventional approach, the figure shows that the handoff users do not have treatment difference from new users. That is why the curves of handoff users overlap with those of new users.

5.4 Summary

This chapter is a critical portion of this thesis. The essential of this chapter is the performance evaluation, and other important things, including performance metrics, related assumptions and required parameters, are also discussed as well.

The six metrics are carefully chosen to use the minimal number of metrics to only evaluate the concerned performance. The average bandwidth utilization indicates whether or not the resources are efficiently utilized. A lower value of this metric may mean that the system wastes the resources. The admission ratio of handoff users and that of new users

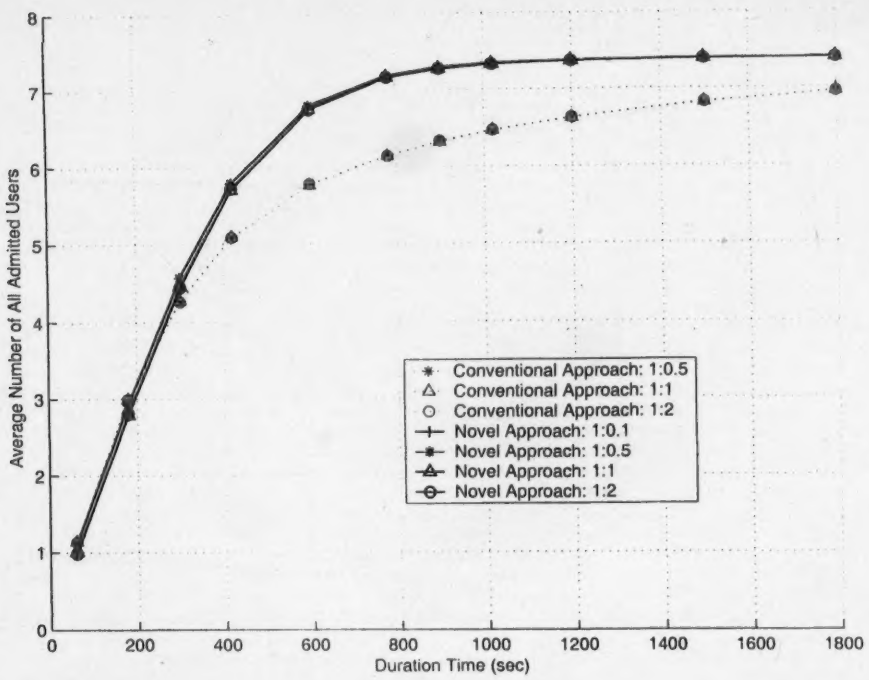

Figure 5.8. COMPARISON OF AVERAGE BANDWIDTH UTILIZATION WITH DIFFERENT RATIO OF TWO TYPES OF USERS

monitor the admission ability of the system. A lower value of this metric means that the system is more likely to reject users. The number of all admitted users denotes the ability of handling multiple users at a time. A lower value of this metric obviously means that the system only can simultaneously deal with fewer users. Thus, the chosen metrics can effectively reflect the system ability of resource utilization, admission and simultaneous handling.

The *ThesisBaseCase* and the *ThesisMyCase* have different sets of parameters, though the two sets overlap. To simplify the simulation with multiple sets of parameters, we allow users to edit a parameter file storing all required parameters. By presenting two examples of the parameter file, the format of the parameter file can be easily understood.

The critical part of this chapter is the performance comparison. By comparing the results derived from a number of experiments, we find that given a set of parameters, our approach is better, or in a very few cases, the same as the conventional approach in all

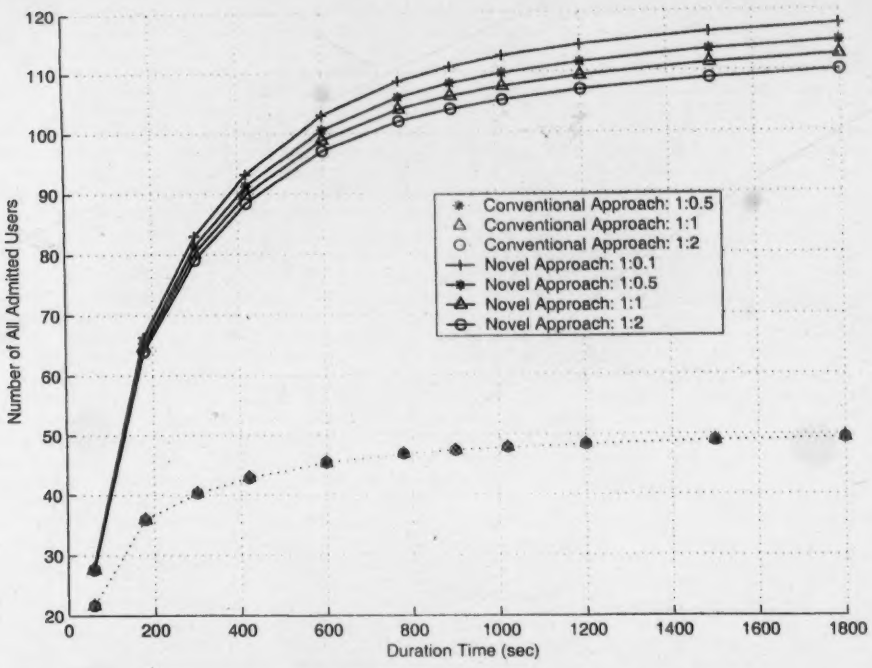

Figure 5.9. COMPARISON OF AVERAGE NUMBER OF ALL ADMITTED USERS WITH DIFFERENT RATIO OF TWO TYPES OF USERS

experiments in terms of the metrics.

The comparison of performance curves of six metrics are listed as follows:

1. The comparison of the metric *the admission ratio of handoff users* shows that:

   (a) Our approach is better than the conventional approach by up to 60%;

   (b) This metric is directly proportional to the interarrival time whereas the metric *the number of admitted handoff users* is inversely proportional to the interarrival time;

   (c) Both this metric and the metric *the number of admitted handoff users* are inversely proportional to the duration time;

   (d) This metric and the metric *the number of admitted handoff users* change more sharply for smaller interarrival time than for larger interarrival time; and

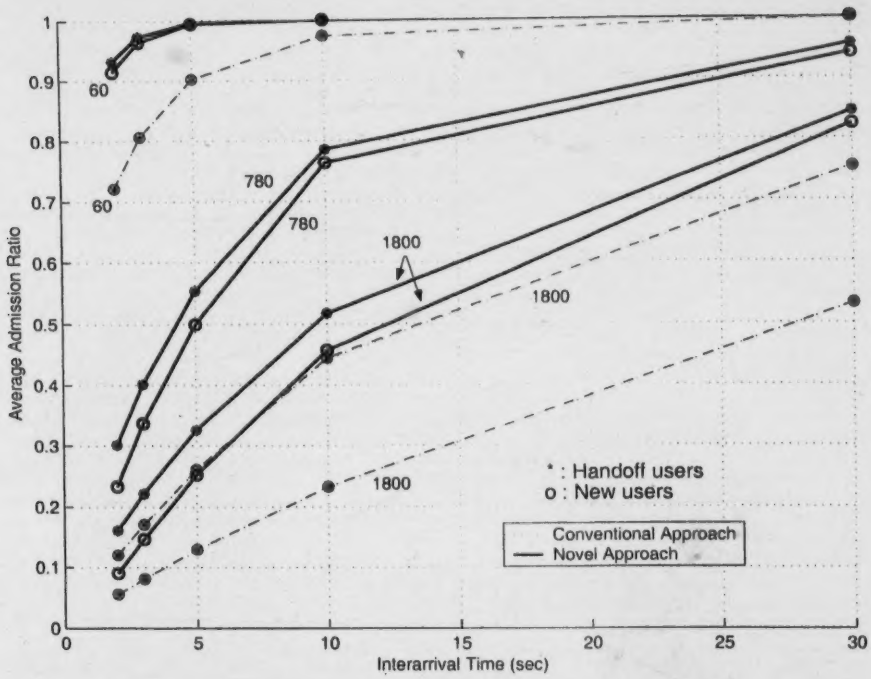

Figure 5.10. AVERAGE ADMISSION RATIO OF HANDOFF USERS VERSUS AVERAGE ADMISSION RATIO OF NEW USERS

(e) This metric and the metric *the number of admitted handoff users* change more sharply for smaller duration time than for larger duration time.

2. The comparison of the metric *the admission ratio of new users* shows that:

   (a) Our approach is better than the conventional approach up to 50%;

   (b) This metric is directly proportional to the interarrival time;

   (c) This metric is inversely proportional to the duration time;

   (d) This metric changes more sharply for smaller interarrival time than for larger interarrival time; and

   (e) This metric changes more sharply for smaller duration time than for larger duration time.

3. The comparison of the metric *the bandwidth utilization* shows that:

(a) Our approach is better than the conventional approach up to 19%;

(b) This metric is inversely proportional to the interarrival time;

(c) This metric is directly proportional to the duration time;

(d) This metric changes more sharply for larger interarrival time than for smaller interarrival time; and

(e) This metric changes more sharply for smaller duration time than for larger duration time.

4. The comparison of the metric *the number of all admitted users* shows that:

   (a) Our approach is better than the conventional approach up to 124%;

   (b) This metric is inversely proportional to the interarrival time;

   (c) This metric is directly proportional to the duration time;

   (d) This metric changes more sharply for smaller interarrival time than for larger interarrival time; and

   (e) This metric changes more sharply for smaller duration time than for larger duration time.

5. The *influence of different ratio of number of handoff users to number of new users* shows that the varied ratio does not result in different performance in conventional approach and it gives fairly small impact on our approach. The possible reason is that the conventional approach does not give higher priority to the handoff users but our approach does. A subtle trend is observed that the larger ratio leads to slightly larger admission ratio in our approach.

6. The *admission ratio of handoff users versus admission ratio of new users* shows that the same conclusion drawn in Figure 5.1 and in our approach, the admission ratio of handoff users is larger than that of new users in general.

Chapter 6

Conclusions and Future Works

In this chapter, we first present some important conclusions and trends we drew from the analysis of simulation results. Then, the future work we plan to extend our approach will be discussed. Finally, the contribution of our thesis is given.

6.1 Contributions

The contributions of this thesis are concluded as follows:

1. A new concept, the *User Profile*, was designed to use the least data to accurately describe the QoS characteristics of connections at the level of the application. It not only contains the information about how much bandwidth is consumed by a user, but also records how many applications are running by this user, the sensitivity of each application and how many resources consumed by each application. Thus, the granularity is greatly improved from the user level to the application level. The *User Profile* enables our approach to be *application-aware*. Clearly knowing information about the bandwidth consumption of every running application enables the system to purposely assign resources to different applications running for different users, instead of fuzzily allocating bandwidth to an *entire* user. That is, the system handles users in an application-oriented way instead of a user-oriented way.

2. The notion of *initial bandwidth* was proposed. This idea is subject to the monthly

fee paid by the subscriber. The carrier is supposed to guarantee that the subscriber can enjoy the initial bandwidth at any time. This notion plays an important role in the bandwidth allocation algorithm and the bandwidth degradation algorithm.

3. We proposed the bandwidth allocation algorithm, which is class-based in consideration of initial bandwidth. Users are allocated bandwidth depending on the request, its class, BUB and the available bandwidth. In the meantime, the *User Profile* is supposed to be updated timely. Moreover, our bandwidth allocation algorithm is *hog-avoidant* due to the constraint of BUB.

4. Since we clearly differentiated RT connections and NRT connections due to their distinctively different characteristics, for the first time the entire bandwidth degradation algorithm was proposed. The basic idea is that due to the high delay tolerance of NRT connections, the bandwidth assigned to NRT connections will be temporarily taken away to admit other requests. This degradation algorithm is class-based and also considers the requirement of initial bandwidth. By using our degradation algorithm, the resources can be employed in a more efficient way and the simulation shows that our approach has higher admission ratio and higher bandwidth utilization than the traditional scheme.

6.2 Conclusions

The conclusions of performance metrics were drawn in Chapter 5. In that chapter, we presented the detailed performance evaluation, ranging from the simulation results to the corresponding data analysis. Some interesting trends of performance metrics were found and their possible reasons were discussed respectively. Here the general conclusion is given as follows:

1. Our approach is better than the conventional approach in all performance metrics (other than a fairly small number of exceptional cases), including the average ad-

mission ratio of handoff users, the average admission ratio of new users, the average number of admitted handoff users, the average bandwidth utilization, and the average number of all admitted users.

2. The admission ratio of handoff users or new users is directly proportional to the interarrival time. In contrast, the number of admitted handoff users, the bandwidth utilization, and the number of all admitted users are all inversely proportional to the interarrival time.

3. Both the admission ratio of handoff users or new users and the number of admitted handoff users are inversely proportional to the duration time. On the contrary, the bandwidth utilization and the number of all admitted users are both directly proportional to the duration time.

4. The absolute value of the first order partial derivative of the admission ratio of handoff users or new users, or the metric the number of admitted handoff users is inversely proportional to the interarrival time. In contrast, the first order partial derivative of the bandwidth utilization or the number of all admitted users is directly proportional to the interarrival time.

5. The absolute value of the first order partial derivative of admission ratio of handoff users or new users, or the metric the number of admitted handoff users is inversely proportional to the duration time. In contrast, the first order partial derivative of the bandwidth utilization or the number of all admitted users is inversely proportional to the duration time.

6. The varied ratio of *influence of different ratio of number of handoff users to number of new users* has no impact on the conventional approach and it gives fairly small impact on our approach.

7. In our approach, the admission ratio of handoff users is larger than that of new users in general.

6.3 Future Work

BUB is Bandwidth Upper Bound, a coefficient indicating how much the MS can use the bandwidth more than the initial bandwidth, i.e. the degree of the greed. It is reasonable to design the system that the higher class has higher BUB. However, due to the tight time for development, we only implemented that all classes have the same BUB. In future work, we will implement different classes with different BUB and explore the effect of varying this parameter.

The bandwidth degradation algorithm apparently admits more users than before in the case when the bandwidth is exhausted. However, the tradeoff is that the system is busier than before if the degradation algorithm is launched. The busyness of the system inevitably influences more or less the system performance, either the user experience or response time. Thus, we need a metric to monitor this inverse influence of the degradation algorithm to see how bad the influence is. In further study, we should find a metric of this kind.

New users are treated fairly similar to handoff users. However, it can be argued that it is much more important to keep handoff calls continued even if more new calls are rejected. We will explore different ways to favor handoff users including not degrading to accept new calls.

Bibliography

[1] CDMA technology. QUALCOMM Inc. [Online]. Available: http://www.cdmatech.com/

[2] (2003) UMTS World. [Online]. Available: http://www.umtsworld.com/technology/wcdma.htm

[3] TD-SCDMA Forum. [Online]. Available: http://www.tdscdma-forum.org/

[4] T. S. Rappaport, *Wireless Communications Principles and Practice*, 2nd ed. Prentice Hall, Dec. 2001.

[5] "Technology Brief Introduction to Gigabit Ethernet," White Paper, Cisco, 1998. [Online]. Available: http://www.cisco.com/warp/public/cc/techno/media/lan/gig/tech/gigbt_tc.pdf

[6] K. Nichols, S. Blake, F. Baker, and D. Black. (1998, Dec.) Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. RFC 2474. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2474.txt

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. (1998, Dec.) An architecture for differentiated service. RFC 2475. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2475.txt

[8] D. Grossman. New terminology and clarifications for DiffServ. RFC 3260. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3260.txt

[9] I. Mahadevan and K. M. Sivalingam, "Architecture and experimental framework for supporting QoS in wireless networks using differentiated services," *Mobile Networks and Applications*, pp. 385–395, Jun. 2001.

[10] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 365–386, Aug. 1995.

[11] R. Guerin, "Queuing-blocking systems with two arrival streams and guarded channels," *IEEE Transaction on Communications*, vol. 36, no. 2, Feb. 1988.

[12] D. Hong and S. S. Rappaport, "Priority oriented channel access for cellular systems serving vehicular and portable radio telephonee," *IEEE Proceedings (UK), Part I, Communications, Speech and Vision*, vol. 136, no. 5, pp. 339–346, Oct. 1989.

[13] C.-J. Chang, T.-T. Su, and Y.-Y. Chiang, "Analysis of a cutoff priority cellular radio system with finite queuing and reneging/dropping," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, Apr. 1994.

[14] B. Li, C. Lin, and S. T. Chanson, "Analysis of a hybrid cutoff priority scheme for multiple classes of traffic in multimedia wireless networks," *Wireless Networks*, vol. 4, no. 4, Aug. 1998.

[15] S.-H. Oh and D.-W. Tcha, "Prioritized channel assignment in a cellular radio network," *IEEE Transaction on Commnunications*, vol. 40, no. 7, pp. 1259–1269, Jul. 1992.

[16] R. Ramjee, R. Nagarajan, and D. F. Towsley, "On optimal call admission control in cellular networks," in *Proc. IEEE INFORCOM'96*, San Francisco, CA, USA, Mar. 1996, pp. 43–50.

[17] D. A. Levine, I. F. Akyildiz, and M. Naghshineh, "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept," *IEEE/ACM Transaction on Networking*, vol. 5, no. 1, Feb. 1997.

[18] C. Oliveiray, J. B. Kimz, and T. Suda, "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks," *IEEE Journal on Selected Areas in Commnunications*, vol. 16, no. 6, Aug. 1998.

[19] B. Li, L. Yin, K. Y. M. Wong, and S. Wu, "An efficient and adaptive bandwidth allocation scheme for mobile wireless networks using an on-line local estimation technique," *Wireless Networks*, pp. 107–116, Jul. 2001.

[20] K. Lee, "Adaptive network support for mobile multimedia," in *Proc. ACM MOBICOM'95*, Berkeley, CA, USA, Nov. 1995.

[21] R. R. F. Liao and A. T. Campbell, "A utility-based approach for quantitative adaptation in wireless packet networks," *Wireless Networks*, vol. 7, pp. 541–557, 2001.

[22] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource reservation protocol," *IEEE Network*, pp. 8–18, Sep. 1993.

[23] G. V. Zaruba and I. Chlamtac, "A prioritized real-time wireless call degradation framework for optimal call mix selection," *Mobile Networks and Applications*, vol. 7, pp. 142–151, 2002.

[24] Y.-B. Lin, A. Noerpel, and D. Harasty, "A non-blocking channel assignment strategy for handoffs," in *Proc. IEEE ICUPC'94*, San Diego, CA, USA, Sep. 1994.

[25] I. Mahadevan and K. M. Sivalingam, "Architecture for QoS guarantees and routing in wireless mobile network," in *Proc. ACM WOWMOM'98*, Dallas, TX, USA, Jun. 1998.

[26] A. K. Tdukdar, B. R. Badrinath, and A. Achary. MRSVP: A reservation protocol for an htegrated services packet network with mobile hosts. Technical Report TR-337. Rutgers University.

[27] ——, "On accommodating mobile hosts in an integrated services packet networks," in *Proc. IEEE INFOCOM'97*, vol. 3, Kobe, Japan, Apr. 1997, pp. 1046–1053.

[28] I. Mahadevan and K. M. Sivalingam, "An experimental architecture for providing QoS guarantees in mobile networks using RSVP," in *Proc. IEEE PIMRC'98*, Boston, MA, USA, Sep. 1998.

[29] S. Singh, "Quality of Service guarantees in mobile computing," *Computer Communications*, vol. 19, pp. 359–371, Apr. 1996.

[30] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 1991.

[31] "Extensible Markup Language (XML) 1.0," White Paper, World Wide Web Consortium (W3C), Jun. 2000. [Online]. Available: http://www.w3.org/TR/REC-xml

[32] G. Malcolm, *Programming Microsoft SQL Server 2000 with XML*, 2nd ed. Microsoft Press, Jun. 2002.

[33] .NET Framework. Microsoft. [Online]. Available: http://msdn.microsoft.com/netframework/

[34] D. S. Platt, *Introducing Microsoft .NET*, 2nd ed. Microsoft Press, Apr. 2002.

[35] E. Meijer and J. Gough, "Technical overview of the Common Language Runtime," Microsoft, Tech. Rep. [Online]. Available: http://research.microsoft.com/~emeijer/Papers/CLR.pdf

[36] (2003) .NET Framework Developer's Guide: Common Language Runtime. Microsoft. [Online]. Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconthecommonlanguageruntime.asp

[37] .NET Framework class library. Microsoft. [Online]. Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/cpref_start.asp

[38] R. Grimes, *Programming With Managed Extensions For Visual C++ .NET*. Microsoft Press, Jul. 2002.

[39] T. Archer and A. Whitechapel, *Inside C#*, 2nd ed. Microsoft Press, Apr. 2002.

[40] A. Hejlsberg, S. Wiltamuth, and P. Golde, "C# language specification," Hewlett-Packard, Intel, and Microsoft, Dec. 2002. [Online]. Available: http://www.ecma-international.org/publications/files/ecma-st/Ecma-334.pdf

[41] *TSG-C Specifications*, The Third Generation Partnership Project 2 Std., 1998 - 2003. [Online]. Available: http://www.2gpp2.com/Public html/specs/index.cfm#tsgc

[42] QUALCOMM Inc. (2001, Nov.) 1xEV: 1x EVolution IS-856 TIA/EIA standard airlink overview. Whitepaper. [Online]. Available: http://www.cdg.org/technology/3g/resource/1xEV_AirlinkOverview_110701.pdf